

HP Client Automation Enterprise

Inventory Manager

for Linux operating systems

Software Version: 7.20

Installation and Configuration Guide

Manufacturing Part Number: None

Document Release Date: July 2008

Software Release Date: July 2008



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

© Copyright 2002-2008 Hewlett-Packard Development Company, L.P.

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.

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

TFTP SERVER

Copyright © 1983, 1993

The Regents of the University of California.

OpenLDAP

Copyright 1999-2001 The OpenLDAP Foundation, Redwood City, California, USA.

Portions Copyright © 1992-1996 Regents of the University of Michigan.

OpenSSL License

Copyright © 1998-2001 The OpenSSLProject.

Original SSLeay License

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

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

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 below lists new features added for this release.

Table 1 New features added for this release

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.
3	7.20	Added instructions for installing the Agent on Thin Client devices.

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	11
	About this Guide	12
	Using this Guide with Core and Satellite Servers	12
	About the Reporting Server	12
	Terminology	13
	HPCA Prerequisites	14
	Hardware/Software	15
	Necessary Skills	15
	With HPCA Products.....	15
	With Web-Based Enterprise Management	15
	Inventory Manager Technology	15
	Web-Based Enterprise Management (WBEM)	15
	Common Information Model (CIM).....	16
	HPCA and WBEM	16
2	Creating the Inventory Manager Environment.....	17
	Inventory Manager Implementation Tasks	18
	SQL or Oracle Prerequisite	18
	Creating the ODBC Database and DSN for Inventory Manager.....	18
	Administrator CSDB Editor	21
	Messaging Server	21
	Reporting Server	21
	Portal (Optional)	22
3	Installing the HPCA agents.....	23
	System Requirements.....	24
	Platform Support.....	24
	Prerequisites	24

Red Hat Linux Prerequisites.....	25
Recommendations.....	26
Installation Methods	27
Including Maintenance Files with the Agent Installation	28
Installing the HPCA agent.....	28
Graphical Installation.....	28
Local Installation.....	28
Remote Installation Setup	31
Non-graphical Installation	36
Installing the HPCA agent to a Thin Client	38
About HPCA Daemons	40
Sample Shell Scripts	40
Troubleshooting the Agent Installation	41
4 The AUDIT Domain	43
The AUDIT Domain	44
AUDIT Domain Defined	45
RIMOPTS Class	47
5 Software and Hardware Auditing.....	51
CIM Schema and Inventory Collection	52
Auditing Types.....	54
File Auditing.....	54
AUDIT.FILE	54
Audit.FILESCAN.....	61
WBEM Auditing.....	62
Manual Scanning Using RIMWBEM	64
NVDCIM.TKD	65
WBEM Object Processing.....	69
Disabling Remnant Configuration Server instances for WBEM Object Processing.....	69
Hardware Auditing	70
6 Successful Auditing	75
Sample Auditing	76

Configuring a Sample Audit	79
Inventory Scan Results.....	81
7 Creating Audit Packages	85
Audit Packages (PACKAGE) Class	86
Using Admin CSDB Editor to Create and Maintain Audit Services	88
Creating UNIX File Audit Methods.....	93
8 Configuring Timers for Audit Collection	97
The Scheduling (TIMER) Class	98
Creating a Timer Instance	103
Specifying Timer Settings.....	105
Specifying ZSCHDEF.....	105
Specifying ZSCHTYPE	106
Specifying ZSCHFREQ.....	106
Specifying ZRSCCMDL.....	107
Specifying ZNOPING.....	107
Connecting the Timer to a Service	107
Audit Execution Configuration.....	108
9 Viewing Inventory from the Reporting Server	113
Accessing the Reporting Server	114
Viewing Audit Information Using the Reporting Server	114
Reporting Views for Inventory Reports	114
Filtering Inventory Reports with Reporting Server	117
Index	119

1 Introduction

At the end of this chapter, you will:

- Understand the components of the HP Client Automation Inventory Manager (Inventory Manager).
- Understand the HP Client Automation (HPCA) prerequisites.
- Realize the skills needed to use the Inventory Manager.
- Be familiar with WBEM and the HPCA agent.
- Be familiar with related HPCA infrastructure components for processing inventory agent data, such as the Messaging Server, Portal, and the Reporting Server.

About this Guide

The Inventory Manager is an agent that discovers configuration information on remote computers. It enables centralized reporting and administration based upon the discovery results. Inventory Manager is installed as part of the HPCA agent.

The Inventory Manager is used with the HP Client Automation Messaging Server (Messaging Server) and HP Client Automation Reporting Server (Reporting Server). Collected inventory data is reported back to the HP Client Automation Configuration Server (Configuration Server) as part of the agent connect process. The Messaging Server handles the routing and posting of data into an inventory database and once in the database, the Reporting Server is used to view inventory data reports.

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

Using this Guide with Core and Satellite Servers

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

About the Reporting Server

As part of the HPCA extended infrastructure, the web-based Reporting Server allows you to query the combined data in existing HPCA databases (including Inventory Manager, Patch Manager, Application Usage Manager and Vulnerability Management) 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.

Refer to the *HP Client Automation Reporting Server Installation and Configuration Guide (Reporting Server Guide)* for more information on how to

install the Reporting Server and create a reporting environment for your HPCA SQL databases.

Terminology

agent computer

An **agent computer** is the computer on the end user's desktop that has the HPCA agent software installed on it.

agent

The **HP Client Automation Agent** is the HPCA software component that is installed on the end user's desktop computer. There are HPCA agents for the Application Manager, the Application Self-service Manager, the Inventory Manager, the Patch Manager, and the OS Manager.

clean machine

A **clean machine** is a desktop computer on which the operating system has just been installed, and no further changes have been made.

Common Information Model (CIM)

The **Common Information Model** is a standardized framework for WBEM. It is an object oriented set of schemas for cross-platform network management. Some of these objects include computer systems, devices (like printers and batteries), controllers (for example, PCI and USB controllers), files, software, etc.

Messaging Server

The Messaging Server is the HPCA infrastructure component that provides a common routing and inter-server data delivery service, especially for report-bound data. When servicing a Configuration Server, the Messaging Server handles the delivery of Inventory, Operations, Patch, and Portal data collected from clients to the appropriate external location.

Portal

The Portal is a Web-based interface used to manage your HPCA infrastructure. The core functionality of the Portal includes: Authentication, Entitlement, Scheduling, Querying, Auditing/Logging, Policy Administration, and instance-level Configuration Server Database Administration. Refer to the *Portal Guide* for more information.

Reporting Server

The Reporting Server is a Web-based interface to the reportable data captured by the HPCA extended infrastructure product suite. It allows you to query the combined data in existing Inventory Manager, Patch Manager, and Application Usage Manager databases and create detailed reports. You have the option of mounting an existing LDAP directory, which allows you to filter your data using your LDAP directory levels.

subscriber

A **subscriber** is the person (end user) who uses HPCA -managed applications on a remote desktop computer (agent computer).

Web-Based Enterprise Management (WBEM)

WBEM enables information such as the amount of RAM in a computer, hard disk capacity, process type, and versions of operating systems to be extracted from computers, routers, switches, and other networked devices.

HPCA Prerequisites

The Inventory Manager requires the following HPCA components:

- Configuration Server 7.20 or higher
- HPCA agent 7.20 or higher
 - Application Manager
- and/or
 - Application Self-service Manager
- Messaging Server 7.20. Refer to the *Messaging Server Guide* for more information on installing or migrating to the Messaging Server, and how the Messaging Server transfers data directly, or indirectly, to HPCA databases.

Hardware/Software

Use the HP Client Automation Administrator CSDB Editor (Admin CSDB Editor) to manipulate the Configuration Server Database (CSDB). The Admin CSDB Editor is part of the HP Client Automation Administrator (Administrator) and is available for Windows platforms. Install the Administrator onto a Windows computer that has access to your CSDB.

Necessary Skills

The following skills are required to use the Inventory Manager.

With HPCA Products

This document assumes that the reader is familiar with the CSDB and with administering HPCA using the Admin CSDB Editor. Refer to the *HP Client Automation Administrator User Guide* for more information.

With Web-Based Enterprise Management

This document assumes that the reader is familiar with Web-Based Enterprise Management (WBEM). To learn more about WBEM go to

<http://www.dmtf.org/spec/wbem.html>

Inventory Manager Technology

While an administrator with little web-based knowledge can use the Inventory Manager with success, it is important to understand some of the technology behind the product. The information provided below gives you a preliminary understanding of the technology behind the Inventory Manager. As indicated in [Necessary Skills](#) above, we recommend that you become familiar with web-based technology.

Web-Based Enterprise Management (WBEM)

Web-Based Enterprise Management (WBEM) is a set of management and Internet standard technologies developed to unify the management of

enterprise computing environments. The Distributed Management Task Force (DMTF) has developed a core set of standards that make up WBEM. The core set includes a data model, the CIM standard, an encoding specification, xmlCIM encoding specification, and a transport mechanism, (CIM Operations over HTTP).

Common Information Model (CIM)

The Common Information Model (CIM) is an object-oriented model that represents and organizes information within a managed environment. This information includes:

- Defining **objects** such as computer systems, devices, controllers, software, files, people, etc.
- Allowing for the definition of **associations** such as describing relationships between object-dependencies, component relationships, and connections.
- Allowing for the definition of **methods** such as input/output parameters and return codes.

By using object-oriented designs and constructs, one of the goals of the CIM model is to consolidate and extend management standards. Some of these management standards include Simple Network Management Protocol (SNMP) and Desktop Management Interface (DMI).

HPCA and WBEM

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

For agent computers with WBEM installed, the Inventory Manager executes an HP proprietary method to query the WBEM namespace.

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

2 Creating the Inventory Manager Environment

At the end of this chapter, you will:

- Be familiar with the tasks needed to set up the HPCA infrastructure and SQL or Oracle database for an HP Client Automation Inventory Manager (Inventory Manager) environment.
- Create a SQL or Oracle database and ODBC DSN connection for the Inventory Manager.
- Have the required infrastructure installed and configured to support an Inventory Manager environment.



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

Inventory Manager Implementation Tasks

To create an HPCA infrastructure environment that supports the use of the Inventory Manager, you will need to complete the following tasks:

- Install the Configuration Server. Refer to the *HP Client Automation Configuration Server, Portal, and Enterprise Manager Getting Started Guide (Getting Started Guide)*.
- Create a SQL or Oracle Database for Inventory Manager data and an ODBC DSN.
- Install the Messaging Server on the same server as the Configuration Server. Select and configure the Data Delivery Agents for Core, Inventory, and WBEM with the ODBC DSN connection needed to post the Inventory Manager data to your SQL or Oracle database. For installation and configuration information, refer to the *Messaging Server Guide*.
- Install the Reporting Server. Refer to the *Reporting Server Guide*.
- Install the Admin CSDB Editor. Refer to the *Administrator Guide*.
- Optional: Install the Portal. Refer to the *Portal Guide*.

SQL or Oracle Prerequisite

Before setting up your environment for the Inventory Manager, you must have already installed the latest version of Microsoft SQL Server 2000 Service Pack 3a or greater.

If using Oracle, the minimum database and driver version is Oracle 9i Release 2, patch set 2 (9.2.0.3).

Creating the ODBC Database and DSN for Inventory Manager

Before installing Inventory Manager agents, create a Microsoft SQL Server or Oracle database. If you do not have security rights to create the database, contact your SQL database administrator.

▶ Client Automation components running on 64-bit systems run in 32-bit emulation mode. Therefore, when using ODBC drivers on 64-bit Windows platforms, you must create the DSN for the ODBC database using 32-bit drivers.

On a Windows 64-bit machine you can access the 32 bit ODBC Data Source Administrator by running

C:\Windows\SysWOW64\odbcad32.exe, to create or modify the DSN's required by our product.

▶ The required size will vary based on the number of managed devices in your environment as well as type of inventory audit information being collected. The procedures below merely reflect recommendations.

To create a Microsoft SQL Inventory Manager database

- 1 Create a database on your Microsoft SQL Server, with the following recommendations:

General tab	Name: HPCAAUDIT (or name of your choice with no blanks or underscores)
-------------	--

Data Files tab	Initial Size: 500 MB Select Autogrow by 20%.
----------------	---

Transaction Log tab	Change initial size: 100 MB
---------------------	-----------------------------

- 2 Use appropriate Microsoft SQL security recommendations for your enterprise.
- 3 On the computer that will be your Configuration Server and co-located Messaging Server, create an ODBC DSN called INVMGR, or name of your choice, pointing to the new INVENTORY database on your SQL Server. If you do not know how to create an ODBC DSN, contact your SQL database administrator.

To create the Oracle database

- 1 Create a tablespace for inventorydata on your Oracle Server with the following recommendations:

Tablespace Name	INVENTORYDATA
Status	Online
Type	Permanent

Datafile	Fully qualified path and name of the datafile such as <code>inventorydata.dbf</code>
Storage	Minimum Size 200 M and Max size unlimited
Extent Management	Locally managed with automatic allocation
Segment Space Management	Automatic
Logging	No

- 2 Create a tablespace for `inventorytemp` with the following recommendations:

Tablespace Name	INVENTORYTEMP
Status	Online
Type	Temporary
Datafile	Fully qualified path and name of the datafile, such as <code>inventorytemp.dbf</code>
Storage	Size 1000 M
Extent Management	Locally managed with automatic allocation
Segment Space Management	Automatic
Logging	No

- 3 Create a user and associate the data and temporary tablespaces to the user with a default profile.

Username	<code>hpcainventory</code>
Password	Create one based on your enterprise's security recommendations.
Default tablespace	INVENTORYDATA
Temporary tablespace	INVENTORYTEMP
Profile	DEFAULT or a PROFILE NAME used for this schema)

- 4 On the computer that will be your Configuration Server and Messaging Server, create an ODBC DSN called `INVMGR`, or name of your choice, pointing to the new `INVENTORY` database on your Oracle Server. If you

do not know how to create an ODBC DSN, contact your Oracle database administrator.



Be careful to ensure that the ODBC driver versions of your Oracle server and your Messaging Server match precisely; the connection to an Oracle database can fail with mismatched ODBC driver versions. For more information, contact your Oracle database administrator.

Administrator CSDB Editor

The HPCA media contains the HPCA Administrator installation, which allows you to install several tools. Use it to install the Admin CSDB Editor. Refer to the *Administrator User Guide* for more information on installing and using the Admin CSDB Editor.



The Admin CSDB Editor is available for Windows platforms only.

Messaging Server

Install the Messaging Server on the Configuration Server. The installation includes the option to install various Data Delivery Agents. Enable these three Data Delivery Agents and configure them with the ODBC DSN needed to post the Inventory Manager-related data to the appropriate back-end Inventory Manager database:

- CORE.DDA
- INVENTORY.DDA
- WBEM.DDA

Refer to the *Messaging Server Guide* for details and requirements.

Reporting Server

The Reporting Server is required to view enhanced reports for Inventory Manager. Review the Release Notes prior to installing. The *Reporting Server Guide* also includes instructions on how to use its flexible features.

Portal (Optional)

The Portal is not required for Inventory Manager. Optionally, it can be used to install the Inventory Manager Agent to groups of devices in your environment remotely. For more information, refer to the *Portal Guide*.

3 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 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, HPCA executables and library files will not be changing with the same frequency as that of your site's user files.

To successfully run HPCA 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	<code>LD_LIBRARY_PATH=/lib:/usr/lib:\$IDMSYS:\$LD_LIBRARY_PATH</code> <code>PATH=/bin:/usr/bin:\$IDMSYS:\$PATH</code>

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 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 `.nvdr` 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/.nvdr
```

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)
- glibclibXaulibXdmc
- libX11
- zlib

Install these files using the appropriate package for your operating system. For example, for RedHat Enterprise Linux 5, these can be installed using the following packages: `ia32el-1.6-13.el5.ia64.rpm`, `glibc-2.5-12.i386.rpm`, `libXau-1.0.1-3.1.i386.rpm`, `libXdmc-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 CSDB. 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 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 HPCA 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 HPCA 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	<code>./install -mode plain</code>	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	<code>./install -mode text</code>	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 36.

- 1 At a command prompt, change your current working directory to the correct Agents platform subdirectory on the installation media.
Example: For Red Hat, 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.

Click **Next**. The Daemons window opens.

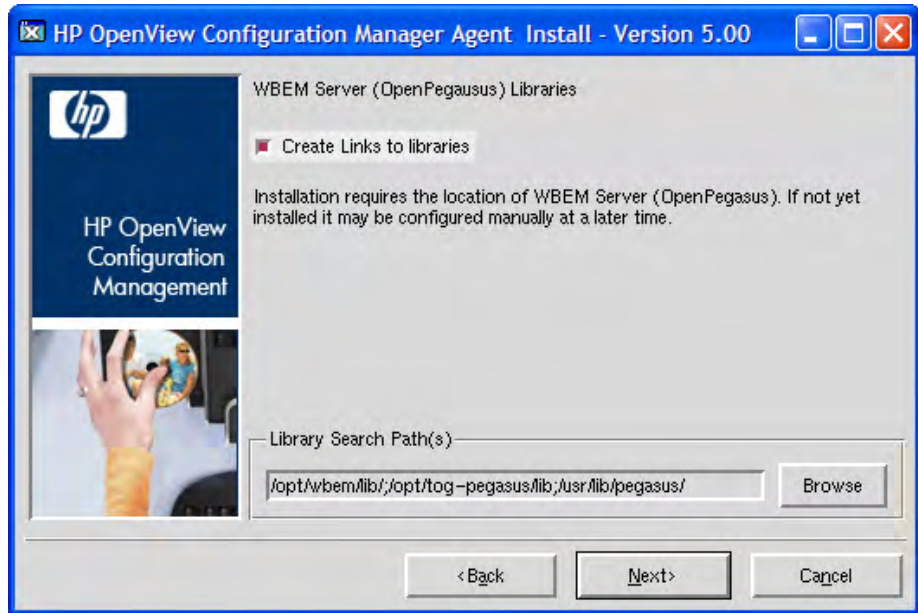
 The next few steps are required for Linux operating systems only. If you are installing to a Mac OS X device, skip to step 11 and select the installation type.

6 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 40 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.

7 Click **Next**. The WbemServer (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.

- 8 Click **Next**. The Select Installation Type window opens.
- 9 Select **Local Install** to install the agent onto a local computer, and then click **Next**. The Agent Location window opens.
- 10 Type the name of the directory where you want to install the agent, or click **Browse** to navigate to it.
- 11 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.

- 12 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.
- 13 Click **Next**. The Log Directory window opens.

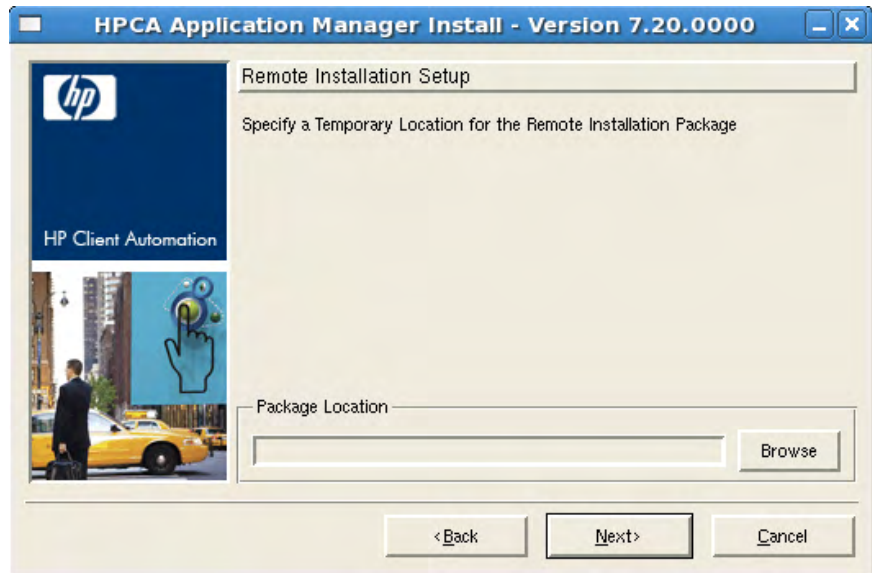
- 14 Type the name of the directory where you would like to store the log files generated by HPCA, or click **Browse** to navigate to it.
- 15 Click **Next**. The Configuration Server IP Address window opens.
- 16 Type the IP address (format: xxx.xxx.xxx.xxx) of the Configuration Server to which the agent will connect. Specify a valid IP address or hostname recognized by the agent workstation.
- 17 Click **Next**. The Configuration Server Port Number window opens.
- 18 Type the Configuration Server's port number (default is 3464).
- 19 Click **Next**. The Package Settings window opens.
- 20 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.
- 21 When you are satisfied with the settings, click **Install** to install the agent with these settings.
- 22 When the installation is complete, click **Finish** to exit the program.
The agent has been successfully installed.

Remote Installation Setup

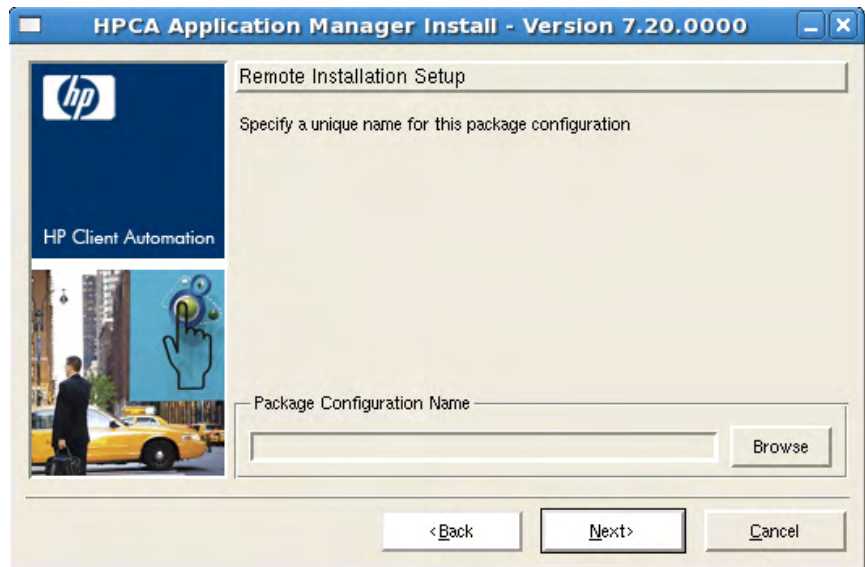
This section describes how to create an agent installation configuration file that can be used to install the Agent in silent mode or to a remote computer.

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](#), below.

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.

Variable	Sample Value	Description
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.
POSTPARAM		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.
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 file name 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 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](#) below describes these.

Table 6 Silent installation command line parameters

Parameter	Example	Description
-cfg	<pre>./install -cfg install.cfg</pre>	The file name specified after <code>-cfg</code> 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 33.
-mode silent	<pre>./install -mode silent -cfg install.cfg</pre>	Installs the agent in silent mode based on the parameters set in the configuration file specified after the <code>-cfg</code> parameter. For information about configuration files, see Customizing the Installation Configuration File on page 33.

Non-graphical Installation

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

To install the 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 28.

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

Example: For Red Hat, 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.



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 a Thin Client

To manage Linux thin client devices, you will need to install the HPCA agent manually using the files provided on the HPCA media.

Installation of the HPCA agent requires minimum free space of 3 MB on the `/mnt` file system. Certain thin client models and related images do not have enough space to install the agent. Currently, models that have only a 32MB flash memory cannot install the agent locally. See notes on running from an NFS share below, and restrictions on using Local Service Boot (LSB) for OS deployment.

To install the Agent on a Linux-based thin client

- 1 Login to the target thin client device as root.
- 2 Create a new directory called `/mnt/opt/OVCM`.
- 3 Copy the contents of `ThinClient.tar` (located on the HPCA media in the `Agents/ThinClient/Linux` directory) to `/mnt/opt/OVCM`.

Depending on your device model, you may have to un-tar these files from `/tmp` or on another machine as some models do not have sufficient disk space to contain both the tar file and its exploded contents (would require approximately 7-8 MB free). After un-tarring, you can delete the `ThinClient.tar`.

- 4 Change the current directory to `/mnt/opt/OVCM` and run the installation by typing:

```
./install -i HPCAS_Server
```

Where `HPCAS_Server` is the hostname or IP address of the HPCA Configuration Server.

The HPCA agent is installed.

- ▶ These devices ship without the ability to contact and register with a DNS server. As such, you may not be able to ping this device. Also, the hosts file created on the factory image has 'localhost.localdomain' as its default hostname, and its (real) assigned hostname as an alias. As a result, the Agent registers the device as localhost.localdomain. You can switch the order in the hosts file to reflect the assigned hostname by placing it first in the list for the 127.0.0.1 entry
- ▶ Management of these devices requires that the BIOS contain a valid serial number and machine UUID (setting asset tag is also recommended). Without these settings, OS deployment may not work properly.

Running the Agent from an NFS Share

If you are using a model that has only 32MB flash memory, you will not be able to install the Agent locally. You will also not be able to use the Local Service Boot option to deploy an OS image and must therefore use PXE for that purpose.

To run the agent remotely from an NFS share:

- 1 Update the install script and modify the MEDIA_RAM_ROOT and INFRA_MEDIA_ROOT variables to point to the NFS directory.
- 2 Create the directory `/mnt/opt/OVCM` and place the install script into this location, and place the rest of the installation package (`thinclient.tar`) into the NFS directory.
- 3 Run install as described in step 4 above.

To remove the Agent from a Linux-based thin client


Use the **uninstall** script to remove the Agent.

- 1 Login to the device as root.
- 2 Go to `/tmp/OVCM/IDMSYS`.
- 3 Type `./uninstall` and hit **Enter**.

The Agent is removed.

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.

4 The AUDIT Domain

At the end of this chapter, you will:

- Understand the AUDIT Domain.
- Understand the Inventory Manager database.

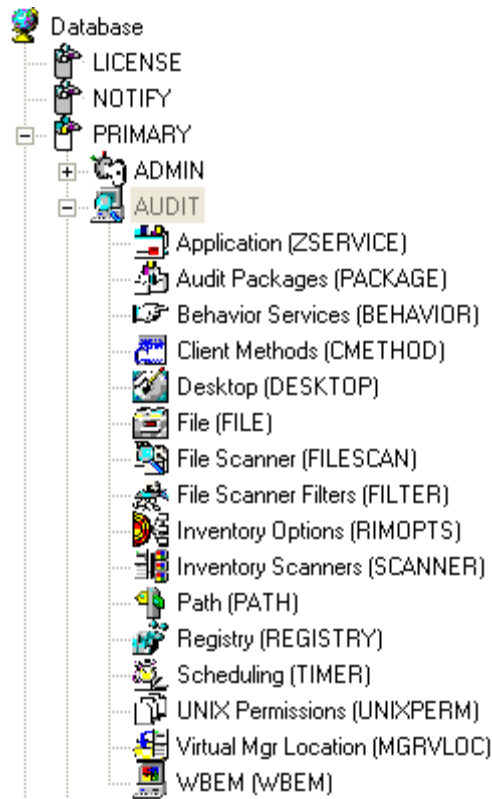
The AUDIT Domain

The AUDIT Domain is located in the PRIMARY File of the HP Client Automation Configuration Server Database (CSDB) and contains the classes required to:

- Configure the tasks needed to collect the inventory information.
- Manage the agent computers' assets.

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

Figure 1 PRIMARY.AUDIT Domain



AUDIT Domain Defined

The AUDIT Domain is structured very much like the SOFTWARE Domain. The following table describes the classes present in the AUDIT Domain.

Table 7 Audit Domain

Class	Description
Audit Application (ZSERVICE)	Sample services distributed with the Inventory Manager. The AUDIT.ZSERVICE instance is connected to a policy instance. A policy instance can be an instance of the Users, Departments, or Workgroups class. It can also be a customer-defined class within the POLICY Domain. Each of the sample ZSERVICE Classes is connected to the PACKAGE instances.
Audit Packages (PACKAGE)	Defines what information to collect and then what actions to take. These packages would contain various audit components. A good example is an audit of running services on a desktop. The AUDIT.ZSERVICE instance must contain a connection to an AUDIT.PACKAGE instance.
Behavior Services (BEHAVIOR)	Defines instances that enable the execution of auditing on the agent. Normally, there is no need to add or modify instances in this class.
Client Methods (CMETHOD)	Used to configure method points for Tcl inventory scans. The base instance of the SCANNER Class is connected to the CMETHOD.INV_FULL instance. This instance can be used for all inventory scans defined in the SCANNER Class.
Desktop (DESKTOP)	This class is reserved for future use.
File (FILE)	Defines file scans, such as auditing system executables.
File Scanner (FILESCAN)	Persistent component class used to configure an inventory scan. Adding File Scanner components to an audit package creates instances of the FILESCAN class.

Class	Description
File Scanner Filters (FILTER)	Persistent component class used to configure an inventory scan. Adding File Scanner Filters components to an audit package creates instances of the FILTER class.
Inventory Options (RIMOPTS)	Contains the attributes that offer options to control an inventory management task.
Inventory Scanners (SCANNER)	Persistent component class that is used to configure an inventory scan. Create instances of the SCANNER class by adding Inventory Scanners components to an audit package.
Path (PATH)	Stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Registry (REGISTRY)	Uses WMI to obtain a Registry scan of a Windows machine. Create instances of the REGISTRY class to run scans of the Windows Registry and obtain a Registry Scan report. Refer to the <i>Registry Class</i> topics for more information.
Scheduling (TIMER)	Contains the instances that enable the HPCA administrator to set a timer on end users' computers. One or multiple auditing services can be processed whenever the timer expires.
UNIX Permissions (UNIXPERM)	Contains UNIX file permission information.
Virtual Mgr Location (MGRVLOC)	Used to specify the initial path for files being transferred to the Configuration Server during a FILE audit.
WBEM (WBEM)	Contains instances that define Inventory Manager scans of WMI classes. These can include any class in the WMI database such as Win32_Services. This example would provide information on Windows NT or Windows 2000 services.

Table 8 FILTER Instances

Instance	Description
NAME	Friendly Name
ACTION	Action Flags: I – Initial (Used for file auditing only [not currently supported]) N – New C – Changed D – Deleted S – Send (upload to Configuration Server) D – Delete (not currently supported) C – Custom (not currently supported)
DIR	Directory to scan.
DEPTH	Number of subdirectory levels to scan Values: -1 root directory and all of its subdirectories 0 root directory only 1 root directory and its files >1 root directory and its files down to the specified depth
INCLUDE	Include globe pattern.
EXCLUDE	Exclude globe pattern.
COMPRESS	Compress [Y/N]
ZRSCVLOC	Name of an instance in the PRIMARY.AUDIT.MGRVLOC class that defines the location to place the uploaded scanned files. Default is RADIA_UPLOAD.

RIMOPTS Class

The RIMOPTS Class is also known as the Inventory Options Class. This class contains the attributes that control an inventory management task. [Table 9](#) on page 48 describes these attributes.

Table 9 RIMOPTS Class

Attribute	Usage
COLLECT	<p>Audit Collection Type by selecting Diff or Full</p> <ul style="list-style-type: none"> • Select Diff to report the difference between the previous information collected for the service and the information collected during the current agent audit. This is the default setting. <p>Note: The first or initial scan of the DIFF setting will be a FULL scan as defined below. All subsequent scans will then be differenced unless the administrator changes the setting to FULL.</p> <ul style="list-style-type: none"> • Select Full to report the information collected for the service during the current agent connect process without differencing against the previous collection for that service.
RUNEXEC	<p>Indicates what actions the Inventory Manager will take upon connection:</p> <ul style="list-style-type: none"> • Select I to invoke collection of information when the service is installed • Select U to invoke collection of information when the service is updated. • Select V to invoke collection of information when the service is verified. <p>The default settings are I and U.</p>
ZSVCTYPE	<p>Contains code that is used internally by the Inventory Manager agent. In all cases, this value should remain I.</p>
NAME	<p>Contains the friendly name of the instance. It is the name displayed for the instance in the tree view of the Admin CSDB Editor.</p>

To apply an option expressed in the RIMOPTS instance to the inventory management task, the RIMOPTS instance must contain a connection to an audit service.

Prior to beginning any tasks using the Inventory Manager, you must enable the drag-and-drop feature for the newly created RIMOPTS Class instances. For more information about editing instances, refer to the *Admin User Guide*.

To enable drag-and-drop connections for RIMOPTS Class instances

- 1 Open the Admin CSDB Editor and go to **PRIMARY** → **ADMIN** → **Name Lists (32) (ZLIST32)** → **CONNECT_** → **CONNECT_ZSERVICE_**
- 2 Double-click **CONNECT_ZSERVICE_TO_RULES**.
- 3 The Editing Instance dialog box opens.
- 4 Set the value of the **ZNAME n** attribute to **RIMOPTS**.

The drag-and-drop feature is now available for all attributes in RIMOPTS.

5 Software and Hardware Auditing

At the end of this chapter, you will:

- Understand file auditing.
- Understand WBEM auditing.
- Understand hardware auditing and the ZCONFIG object.

CIM Schema and Inventory Collection

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

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

For a description of the CIM schema classes used, see [Table 10](#) below.

Table 10 CIM classes

CIM Class	Description
CIM_SCSIController	Subclass of the CIM_Controller used to represent SCSI controllers.
CIM_ResidesOnExtent	Subclass of CIM_Dependency. This is an association between the logical volume and the file system on the logical volume.
CIM_Processor	Used to represent computer processor information.
CIM_ParallelController	Subclass of CIM_Controller used to represent parallel controllers.
CIM_NFS	Used to represent general information about NFS mounted file systems.
CIM_MediaPresent	Used to represent relationship with the MediaAccessDevice. Represents logical volume or volume group and one of the disks it resides on.
CIM_LogicalDiskBasedOnVolume	Subclass of LogicalDiskBasedOnExtent used to represent the relationship between logical volume and its volume group.
CIM_LogicalDisk	Used to represent general information about the logical volume.

CIM Class	Description
CIM_IDEController	Subclass of CIM_Controller used to represent IDE controllers, including ATA and ATAPI controllers.
CIM_EthernetAdapter	Used to represent capabilities of the Ethernet card.
CIM_DiskDrive	Subclass of CIM_MediaAccessDevice, includes all hard disk drives, non-removable and removable. Models the reader/writer properties of disk drives.
CIM_Directory	Used for exported directory.
CIM_DVDDrive	Subclass of CIM_MediaAccessDevice includes all of the types of DVD reader and writer drives.
CIM_CDROMDrive	Subclass of CIM_MediaAccessDevice includes CDROM reader and writer drives.
CIM_Service	Used to represent general information about NFS client/server service.
CIM_SCSIInterface	Subclass of CIM_ControlledBy. Represents unique data from the relationship between the controller and the device.
CIM_UnixLocalFileSystem	Used to represent UNIX specific information about the local file system.
CIM_UnixComputerFileSystem	Used to represent general information about the computer.
CIM_StorageVolume	Used to represent the hand-off point between providers or the result of a redundancy.
CIM_UnixOperatingSystem	Used to represent general information about the UNIX operating system. General information about the volume groups.

CIM Class	Description
CIM_SoftwareElement	Used to represent the SVR4 packages or filesets.
CIM_Export	Used to represent an association between a LocalFileSystem and its directories indicating that the specified directories are available for mount. When exporting an entire FileSystem, the directory should reference the topmost directory of the FileSystem.

For more information about the CIM schema 2.6 visit the DMTF Web site:

<http://www.dmtf.org/>.

Auditing Types

When configuring your audits, the administrator should understand exactly what types of things can be audited and what the expected results from an audit will comprise.

The Inventory Manager allows for three types of audits:

- File auditing
- WBEM auditing
- Hardware auditing

File Auditing

AUDIT.FILE

The AUDIT.FILE class instances in an audit package control the auditing function for files on the agent computer. The RIMFSCAN and the RIMDIFF methods on the agent computer perform the actual file auditing operations by specifying what files to look for. There can be one or more AUDIT.FILE instances in an audit package. Each AUDIT.FILE instance can specify a scan for one or more files.

The following table summarizes the attributes in an `AUDIT.FILE` class instance and their effects on the `RIMFSCAN` method.

Table 11 `AUDIT.FILE` Class Instances

Attribute	Description Examples
SCANFOR	Indicate a fully qualified path and file name to search for. Wildcards are permitted.
ACTION	<p>The <code>RIMDIFF</code> method performs actions on the files discovered on the user's computer during the agent connect.</p> <ul style="list-style-type: none"> • Y configures <code>RIMDIFF</code> to perform the action. • N configures <code>RIMDIFF</code> to not perform the action. <p>The first four flags determine when to report that the files were found: Report on: Initial, New, Changed, Deleted</p> <ul style="list-style-type: none"> • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan. • Deleted means that the file was found during the previous scan. The file is not present for the current scan. <p>The last three flags control the actions to take on the files detected during the current scan. Action to take on discovery: Send, Delete, Custom</p> <ul style="list-style-type: none"> • Send means to send the files to the Configuration Server and store them in the location indicated by the <code>ZRSCVLOC</code> attribute (see <code>ZRSCVLOC</code> in this table). • Delete means to delete the files from the user's computer. • Custom means to execute the method indicated in the <code>CUSTOM</code> attribute. <p>YYYYNYN – Report whenever encountered and delete the files.</p>

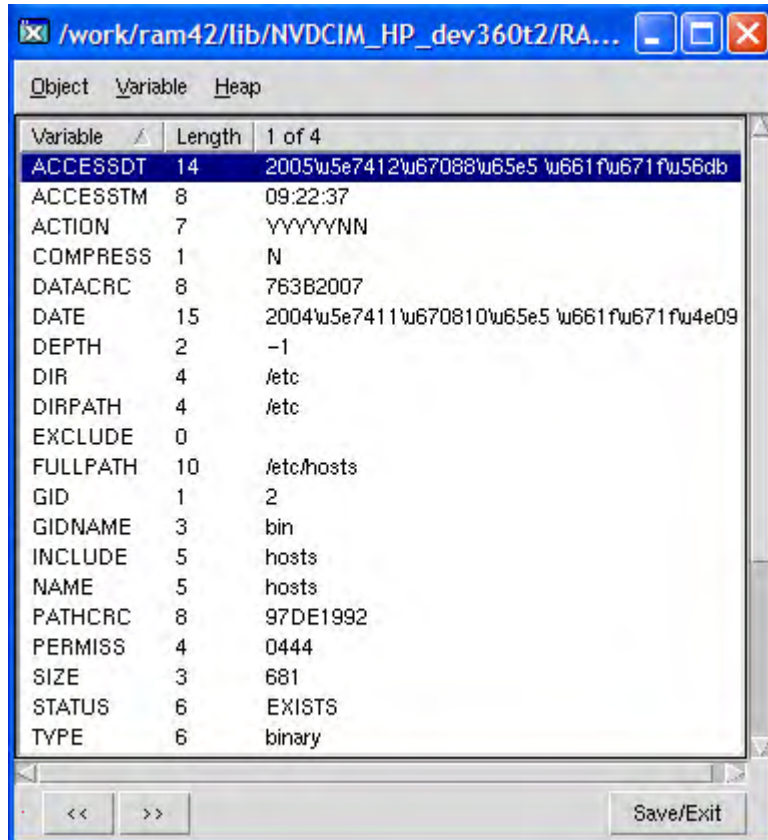
Attribute	Description Examples
	<p>NNYYNNN – Report when changed or deleted and take no action.</p> <p>NYYNYYN – Report when the files are new or changed. Then send and delete the files.</p>
OUTPUT	Output object name.
TYPE	<p>Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM. File.</p>
GROUP	<p>Optional way to identify a set of scan results. This maybe useful for querying and reporting on the audited files from the database where audit results can be stored. Games, MPEGs.</p>
ZVERINFO	<p>Collect extended information.</p> <ul style="list-style-type: none"> • Set the value to 1 to collect more information for a file. • Set the value to 0 to not collect more information. <p>In order for this data to be collected, the associated attribute must exist in the AUDIT.FILE class template. You can limit the scan to only those files that have some particular values in their extended information. You do so by supplying a value (either 1 or 0) for any of the associated attributes in an AUDIT.FILE instance. This causes the scan to be filtered. Only those files whose extended information data element contains the value you specify in its associated attribute will be scanned.</p> <p>Extended file information consists of one ore more of the following data elements. The associated attribute name for the data element is in parentheses:</p> <p>(VENDOR) The seller of the file/product</p> <p>(PRODUCT) The name of the item for which the file is a part.</p> <p>(PRODVERS) The version of the product which the file is a part.</p> <p>(ORGNAME)</p>

Attribute	Description Examples
	<p>The name of the organization. (INTERNAL)</p> <p>The internal data element encoded in the file. (VERSION)</p> <p>The version of the file. (LANGUAGE)</p> <p>The language of the file.</p>
ZRSCSTYP	Server file type. This can be either Binary or Text. The administrator does not set this.
ZRSCMFIL	Manager directory location.
ZRSCVLOC	The location on the Configuration Server where the files are stored because of the Send Action (see ACTION in this table). This variable needs to be configured when sending a file back to the Configuration Server. The variable should contain the name of the MGRVLOC instance that will be used to resolve the location to store the uploaded file.
ZRSCMMEM	PDS member name. This field is optional.
PRODUCT	<p>The product name.</p> <p>See ZVERINFO on page 56 for more detail.</p>
PRODVERS	<p>The product version.</p> <p>See ZVERINFO on page 56 for more detail.</p>
ORGNAME	<p>The organization name.</p> <p>See ZVERINFO on page 56 for more detail.</p>
INTERNAL	<p>The internal data element encoded in the file.</p> <p>See ZVERINFO on page 56 for more detail.</p>
VERSION	<p>The version of the file.</p> <p>See ZVERINFO on page 56 for more detail.</p>
LANGUAGE	<p>The language of the file.</p> <p>See ZVERINFO on page 56 for more detail.</p>
VENDOR	<p>The product vendor.</p> <p>See ZVERINFO on page 56 for more detail.</p>

Attribute	Description Examples
ZRSCCRC	Resource CRC.
ZCRCINFO	Collect file CRC.
ZRSCOBJN	Persistent object name.
ZRSCPADM	Administrator ID.
ZRSCSRC	Resource Source, i.e. Publisher.
ZINIT	Not applicable at this time.
NAME	Not applicable at this time.
LOCATION	Not applicable at this time.
ZMD5INFO	Set to Y to collect MD5 info. This is a 32-character value that can be used to uniquely identify a file based on its contents.

Use the Admin Agent Explorer to view the FILEPREV object results as shown in [Figure 2](#) on page 59.

Figure 2 FILEPREV object created with RIMFSCAN



The screenshot shows a debugger window titled "/work/ram42/lib/NVDCIM_HP_dev360t2/RA...". The window displays a table of variables for the FILEPREV object. The table has three columns: Variable, Length, and Heap. The first row is highlighted in blue. The variables listed are ACCESSDT, ACCESSTM, ACTION, COMPRESS, DATACRC, DATE, DEPTH, DIR, DIRPATH, EXCLUDE, FULLPATH, GID, GIDNAME, INCLUDE, NAME, PATHCRC, PERMISS, SIZE, STATUS, and TYPE.

Variable	Length	Heap
ACCESSDT	14	2005\u5e7412\u67088\u65e5 \u661f\u671f\u56db
ACCESSTM	8	09:22:37
ACTION	7	YYYYNN
COMPRESS	1	N
DATACRC	8	763B2007
DATE	15	2004\u5e7411\u670810\u65e5 \u661f\u671f\u56db
DEPTH	2	-1
DIR	4	/etc
DIRPATH	4	/etc
EXCLUDE	0	
FULLPATH	10	/etc/hosts
GID	1	2
GIDNAME	3	bin
INCLUDE	5	hosts
NAME	5	hosts
PATHCRC	8	97DE1992
PERMISS	4	0444
SIZE	3	681
STATUS	6	EXISTS
TYPE	6	binary

The FILEPREV object contains one heap for each file discovered during the scan for the audit service. It contains the attributes from the AUDIT.FILE class instance that controlled the scan, as described above. It also contains the following attributes:

Table 12 FILEPREV Object

Attribute	Description
ACTION	Action flags. First four flags determine when to report. Y – ignored Y – New file Y – File changed since last scan Y – Ignored Last three flags control action to be taken. Y – send the file to RCS Y – ignored Y – ignored
ACCESSDT	The date of the most recent access of this file.
ACCESSTM	The time of the most recent access of this file.
COMPRESS	Compression setting.
DATA CRC	Data CRC
DATE	The date of the most recent modification to this file.
DIR	System drive location of the file.
DIRPATH	The directory path of the file.
EXCLUDE	Parameter to exclude.
FULLPATH	Fully qualified path and file name of the file.
GID	Unix group ID of file owner.
GIDNAME	Unix group name of file owner.
INCLUDE	Parameter to include.
NAME	File name.
PATHCRC	A unique number that indicates the CRC path used for differencing.
PERMISS	4-digit octal value for file permissions.
SIZE	File size in bytes.
TIME	The time of the most recent modification to this file.
TYPE	File type. Can be directory, LINK, or binary.

Attribute	Description
UID	UNIX ID of file owner.
UIDNAME	Username of the file owner.
ZOBJDATE	Date
ZOBJPCLAS	Class
ZOBJCID	Object Child ID
ZOBJPNAM	Unique Name
ZOBJTIME	Time
ZRSCVLOC	Location

Audit.FILESCAN

- UNIX file auditing using `filescan.tkd` is supported for legacy purposes only. New installations of the Inventory Manager should use RIMFSCAN and RIMDIFF, described in the section `AUDIT.FILE`, above.
- Resolution of the `filescan.tkd` service is not supported over SSL.

The `AUDIT.FILESCAN` class instances in an audit package control the auditing function for files on the agent computer. The `filescan.tkd` methods on the agent computer perform the actual file auditing operations by specifying what files to look for. There can be one or more `AUDIT.FILESCAN` instances in an audit package. Each `AUDIT.FILESCAN` instance can specify a scan for one or more files.

See [Inventory Scan Results](#) on page 81 for more information on the `filescan.tkd` methods.

The following table summarizes the attributes in an `AUDIT.FILESCAN` class instance and their affects on the `filescan.tkd` method.

Table 13 `AUDIT.FILESCAN` Class Instances

Attribute	Description
NAME	Friendly name.

Attribute	Description
DIFF	Specifies if differencing is to be done or not. If DIFF = Y, then the information from the scanned files will be compared with the information from the previous file scan.
OUTPUT	Specifies the prefix to be used for the object names created. If OUTPUT=FILE, then FILEAUDIT, FILEPREV objects will be created on the agent computer.

WBEM Auditing

Use the RIMWBEM method to query the WBEM namespaces to retrieve information about how a system's hardware and software is used. The RIMWBEM method constructs a query from the information contained in an instance of the AUDIT.WBEM Class. WBEM has a query engine that processes the query statement and returns the query results to RIMWBEM. There is one heap in the query result object for every discovered instance.

An AUDIT.WBEM class instance defines a query into the WBEM namespace.

The following table describes the attributes of the AUDIT.WBEM instance.

Table 14 AUDIT.WBEM Instance

Attribute Name	Description
ACTION	<p>RIMDIFF method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect.</p> <ul style="list-style-type: none"> • Y configures RIMDIFF to perform the reporting action. • N configures RIMDIFF to not perform the reporting action. <p>The first four flags determine <i>when</i> to report that the WBEM namespace instance was found: Report on: Initial, New, Changed, Deleted, Scan, Delete, Custom</p> <ul style="list-style-type: none"> • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present

Attribute Name	Description
	<p>during the previous scan and is different from the file found during the current scan.</p> <ul style="list-style-type: none"> • Deleted means that the file was found during the previous scan. The file is not present for the current scan. • Scan means that the file was found during the current scan. • Delete means that the file was found during the previous scan. The file is not present for the current scan. • Custom means that the file was found during a custom scan. <p>The last three flags are not applicable to WBEM audits.</p>
NAMESPACE	Name of the WBEM namespace to query or HARDWARE .
CLASS	Name of the WBEM Class to query or HARDWARE .
PROPERTY	<p>Specify one or more property names to be queried and reported. Use commas to separate more than one property name.</p> <p>If this attribute is blank, all properties in the class will be queried and reported.</p>
CONDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.
NAME	Friendly name for this instance. This name will appear in the Admin CSDB Editor tree view to identify this instance.



When the keyword **HARDWARE** is used in the **NAMESPACE** and/or **CLASS** attributes of **AUDIT.WBEM**, hardware information is collected. This information is essentially the same as the **ZCONFIG** object.

The agent stores the results of a WBEM scan in a WBEM object. This object can be found in the service node of the client object tree. The results are also sent to the Configuration Server.

In addition to the attributes described above, the WBEM object also contains the following:

Table 15 WBEM object attributes in the agent

Attribute	Description
ZOBJCID	Object child ID.
ZOBJCLAS	Targeted class for the audit such as ZRSOURCE or ZSERVICE.
ZOBJCRC	CRC of all persistent and transient objects under the current node.
ZOBJDATE	Last date under the current node.
ZOBJDOMN	Domain name of the object.
ZOBJID	Object ID of the instance used to obtain information from the Resource file.
ZOBJNAME	Instance name of the object.
ZOBJPCLS	Parent class name.
ZOBJPID	Parent class ID.
ZOBJRCRC	Resource CRC maintained by the Configuration Server.
ZOBJRSIZ	Resource size maintained by the Configuration Server.
ZOBJTIME	Latest time under the current node.
ZRSCSRC	Name of the program promoted the resource.

Manual Scanning Using RIMWBEM

RIMWBEM can be run from the command line to manually scan for a particular WBEM Class using the following syntax:

```
./rimwbem class=CIM_ComputerSystem
```

The example above will scan for CIM_ComputerSystem. Replace this CIM provider name with any WBEM Class information for which you want to manually scan. After the scan, check `$IDMSYS/log/rimwbem.log` and `$IDMSYS/lib/WBEMCURR.EDM` for the results.

To verify the results of the scan, run a custom query using a CIM navigator program (for example, CimNavigator).

NVDCIM.TKD

► UNIX WBEM auditing using `nvdcim.tkd` is supported for legacy purposes only. New installations of the Inventory Manager should use RIMWBEM, described in the section WBEM Auditing, above.

The `nvdcim.tkd` method is used to query the WBEM namespaces to retrieve information about a system's hardware and software. The method constructs a query from the information contained in an instance of the AUDIT.WBEM Class. WBEM has a query engine that processes the query statement and returns the query results to `nvdcim.tkd`. There is one heap in the query result object for every discovered instance.

An AUDIT.WBEM class instance defines a query into the WBEM namespace.

Figure 3 **AUDIT.WBEM Class instances**

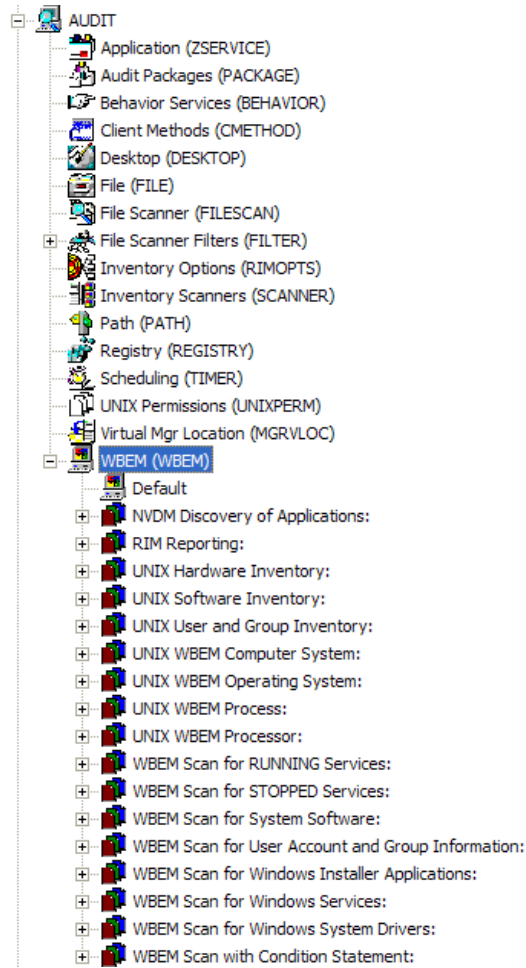


Table 16 on page 67 describes the attributes of the AUDIT.WBEM instance.

Table 16 AUDIT.WBEM Instance

Attribute	Description
ACTION	<p>The <code>filescan.tkd</code> method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect.</p> <ul style="list-style-type: none">• Y configures <code>filescan.tkd</code> to perform the reporting action.• N configures <code>filescan.tkd</code> to not perform the reporting action. <p>The first four flags determine <i>when</i> to report that the WBEM namespace instance was found: Report on: Initial, New, Changed, Deleted</p> <ul style="list-style-type: none">• Initial means that the file was found during the first scan of the agent computer.• New means that the file was found during the current scan. The file was not present during the previous scan.• Changed means that the file was present during the previous scan and is different from the file found during the current scan.• Deleted means that the file was found during the previous scan. The file is not present for the current scan. <p>The last three flags are not applicable to WBEM audits.</p>
NAMESPACE	Name of the WBEM namespace to query or HARDWARE .
CLASS	Name of the WBEM Class to query or HARDWARE .
PROPERTY	<p>Specify one or more property names to be queried and reported. Use commas to separate more than one property name.</p> <p>If this attribute is blank, all properties in the class will be queried and reported.</p>
CONDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.

Attribute	Description
NAME	Friendly name for this instance. This name will appear in the Admin CSDB Editor tree view to identify this instance.



When the keyword **HARDWARE** is used in the **NAMESPACE** and/or **CLASS** attributes of **AUDIT.WBEM**, hardware information is collected. This information is essentially the same as the **ZCONFIG** object.

The agent stores the results of a **WBEM** scan in a **WBEM** object. This object can be found in the service node of the client object tree. The results are also sent to the Configuration Server.

The **WEBM** object contains more attributes described in [Table 17](#) below.

Table 17 WBEM object attributes

Attribute	Description
ZOBJCID	Object child ID.
ZOBJCLAS	The targeted class for the audit such as ZRSOURCE or ZSERVICE .
ZOBJCNUM	Number of children under current instance.
ZOBJCRC	The CRC of all persistent and transient objects under the current node.
ZOBJDATE	The last date under the current node.
ZOBJDOMN	The domain name of the object.
ZOBJID	The object ID of the instance used to obtain information from the Resource file.
ZOBJNAME	The instance name of the object.
ZOBJPCLS	The parent class name.
ZOBJPID	The parent class ID.
ZOBJRCRC	The resource CRC maintained by the Configuration Server.
ZOBJRSIZ	The resource size maintained by the Configuration Server.

Attribute	Description
ZOBJTIME	The latest time under the current node.
ZRSCSRC	The name of the program promoted the resource.
ZUNUSED1	For future use.

WBEM Object Processing

When the Inventory Manager agent sends a WBEMAUDT object to the Configuration Server, processing is defined as follows:

- 1 When a WBEMAUDT object is found, the Configuration Server ZTASKEND calls QMSG.EXE.
- 2 QMSG.EXE places the WBEMAUDT objects into the Configuration Server's \data\wbem directory, or message queue.
- 3 The Messaging Server includes a WBEM Data Delivery Agent (WBEM.DDA) that monitors this \data\wbem message queue and processes the WBEM objects.
- 4 The WBEM.DDA is usually configured to post the WBEM objects directly to an ODBC-compliant Inventory Manager database, or, it may be configured to first forward the WBEM objects to another Messaging Server located closer to the database. In the later case, the receiving Messaging Server posts the WBEM data to the Inventory ODBC-compliant database.
- 5 Once posted to the Inventory database, the new WBEM information is immediately available for query and reporting purposes through the Reporting Server.

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

Disabling Remnant Configuration Server instances for WBEM Object Processing

Inventory Manager no longer supports processing WBEM objects using these instances in the Configuration Server database:

- SYSTEM.PROCESS.WBEMAUDT
- SYSTEM.ZMETHOD.POST_WBEM

If these remnant instances exist or were imported into your Configuration Server database, you must disable any configurations within them in order to ensure successful WBEM object processing.

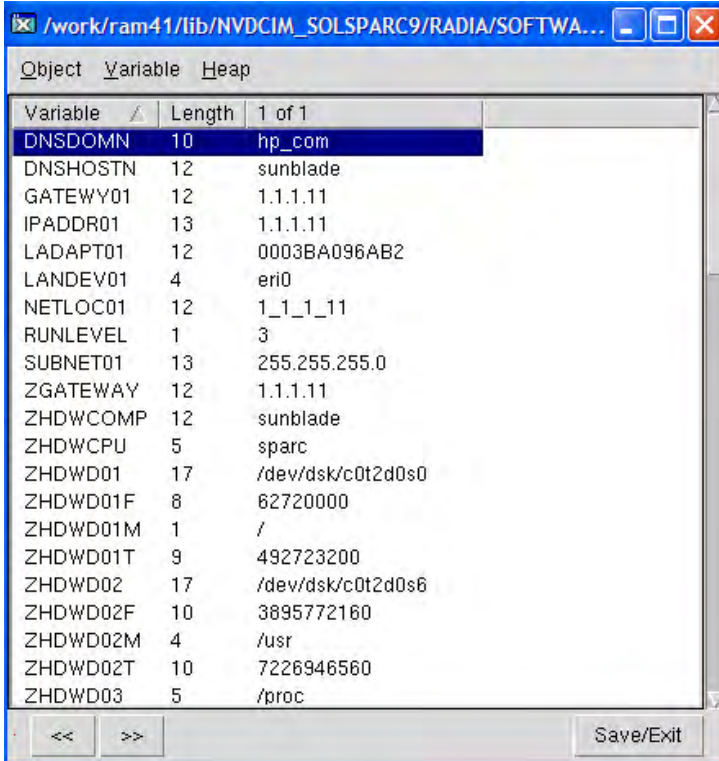
Edit SYSTEM.PROCESS.WBEMAUDT and remove any connection to the SYSTEM.ZMETHOD.POST_WBEM instance.

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

Hardware Auditing

Each time an agent connects to the Configuration Server, information about the subscriber's hardware configuration is stored in the ZCONFIG object. The ZCONFIG object is calculated and stored in the application service directory of the agent's object directory tree.

Figure 4 ZCONFIG object in Admin Agent Explorer

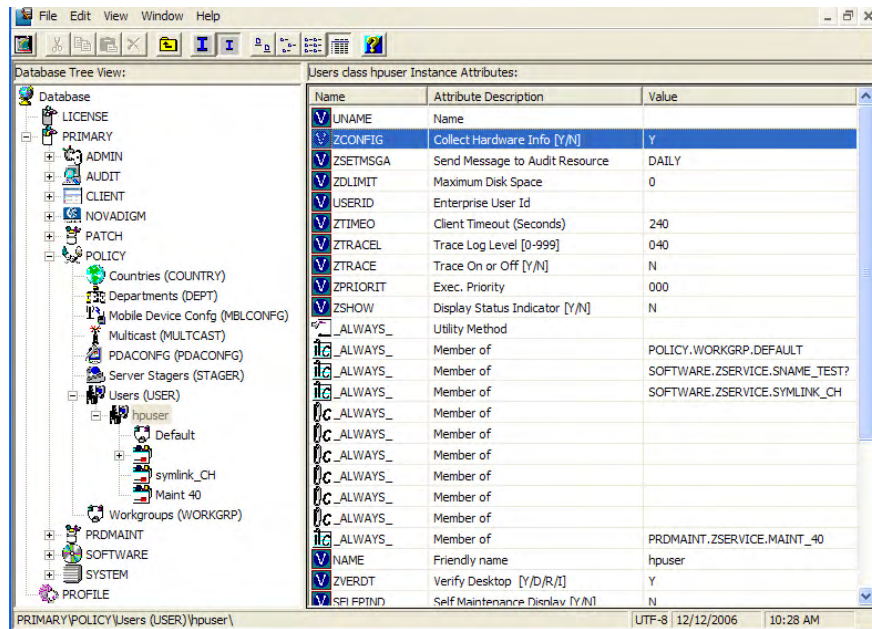


Variable	Length	1 of 1
DNSDOMN	10	hp_com
DNSHOSTN	12	sunblade
GATEWY01	12	1.1.1.11
IPADDR01	13	1.1.1.11
LADAPT01	12	0003BA096AB2
LANDEV01	4	eri0
NETLOC01	12	1_1_1_11
RUNLEVEL	1	3
SUBNET01	13	255.255.255.0
ZGATEWAY	12	1.1.1.11
ZHDWCOMP	12	sunblade
ZHDWCPU	5	sparc
ZHDWD01	17	/dev/dsk/c0t2d0s0
ZHDWD01F	8	62720000
ZHDWD01M	1	/
ZHDWD01T	9	492723200
ZHDWD02	17	/dev/dsk/c0t2d0s6
ZHDWD02F	10	3895772160
ZHDWD02M	4	/usr
ZHDWD02T	10	7226946560
ZHDWD03	5	/proc

A separate ZCONFIG object is calculated and stored for each service installed or updated during the agent connect process.

To force the transfer of the hardware information, the ZCONFIG variable *must* be set to Y in the POLICY.USER class (see figure below). To change this, use the Admin CSDB Editor, which is available for 32-bit Windows platforms.

Figure 5 POLICY.USER class – ZCONFIG variable



The ZCONFIG object stores information about the agent computer's hardware.

Figure 6 Sample ZCONFIG object in Admin Agent Explorer

The screenshot shows a window titled "/work/ram42/lib/NVDCIM_HP_dev360t2/RA...". Below the title bar is a menu bar with "Object", "Variable", and "Heap". The main area contains a table with the following data:

Variable	Length	1 of 1
ZHDWCOMP	8	dev360t2
ZHDWCPU	8	9000/800
ZHDWD00	15	/dev/vg00/vol3
ZHDWD00F	9	439689216
ZHDWD00M	1	/
ZHDWD00T	9	528482304
ZHDWD00U	8	85360640
ZHDWD01	15	/dev/vg00/vol1
ZHDWD01F	8	43057152
ZHDWD01M	6	/stand
ZHDWD01T	8	88457216
ZHDWD01U	8	36552704
ZHDWD02	15	/dev/vg01/vol1
ZHDWD02F	11	29607206912
ZHDWD02M	5	/work
ZHDWD02T	11	36410753024
ZHDWD02U	10	6588162048
ZHDWD03	15	/dev/vg00/vol8
ZHDWD03F	10	3186696192
ZHDWD03M	4	/var
ZHDWD03T	10	4194304000

At the bottom of the window, there are navigation buttons "<<" and ">>" and a "Save/Exit" button.

The ZCONFIG object stores hardware information discovered by the agent's standard hardware auditing method. Certain types of hardware can occur multiple times. The ZCONFIG object automatically expands to allow more information to be stored.

The following table describes the variables that are stored in a sample ZCONFIG object.

Table 18 Attributes in a Sample ZCONFIG

Attribute	Description	Example
DESCRIPT	<i>Internal use only</i>	Processing Client Request for &ZCUROBJ
IPADDR01	IP address of network adapter 1	1.1.1.99

Attribute	Description	Example
LADAPT01	LAN Adapter 1	02608C2CBDCE
LANNUM	LAN Number	1643292
OSREV	Operating System revision number	4
OSVER	Operating System Version	3
ZHDWCPU	CPU Type	000019131C00
ZHDWD00	Drive Name for Drive 00	/dev/hd4
ZHDWD00F	Current free space on drive 00	7028736
ZHDWD00M	Drive 00 mount	/
ZHDWD00T	Total space for drive 00	25165824
ZHDWD01	Drive name for drive 01	/dev/hd2
ZHDWD01F	Current free space on drive 01	15859712
ZHDWD01M	Drive 01 mount	/usr
ZHDWD01T	Total space for drive 01	1577058304
ZHDWD02	Drive name for drive 02	/dev/hd9var
ZHDWD02F	Current free space on drive 02	2973696
ZHDWD02M	Drive 02 mount	/var
ZHDWD02T	Total space for drive 02	16777216
ZHDWD03	Drive name for drive 03	/dev/hd3
ZHDWD03F	Current free space on drive 03	28729344
ZHDWD03M	Drive 03 mount	/tmp
ZHDWD03T	Total space on drive 03	41943040
ZHDWDNUM	Number of drive letters assigned	9
ZHDWIPAD	IP address	&(IPADDR01)
ZHDWLANA	LAN Adapter	&(LADAPT01)
ZHDWMEM	Total physical memory (RAM)	65536
ZHDWOS	Operating system and version	Red Hat
ZHDWXHID	Host ID	0x1010163

Attribute	Description	Example
ZHDWXHN	Host name	linuxdemo
ZOBJRRC	Resolution return code	000
ZOBJRSTY	Resolution type	C
ZSRCCLAS	Service class	ZCONFIG
ZSRCCRC	Service CRC	8B37472C
ZSRCDATE	Service date	20001211
ZSRCDOMN	Service domain	SYSTEMX
ZSRCNAME	Service name	HARDWARE_SCAN
ZSRCPID	Service parent ID	0000000000
ZSRCTIME	Service time	11:52:59
ZUSERID	User ID	jdoe

Whenever an agent connects to the Configuration Server, certain subscriber hardware information is automatically forwarded to the Inventory Manager ODBC database as part of the Messaging Server processing of CORE objects. Use the Reporting Server to view hardware information.

6 Successful Auditing

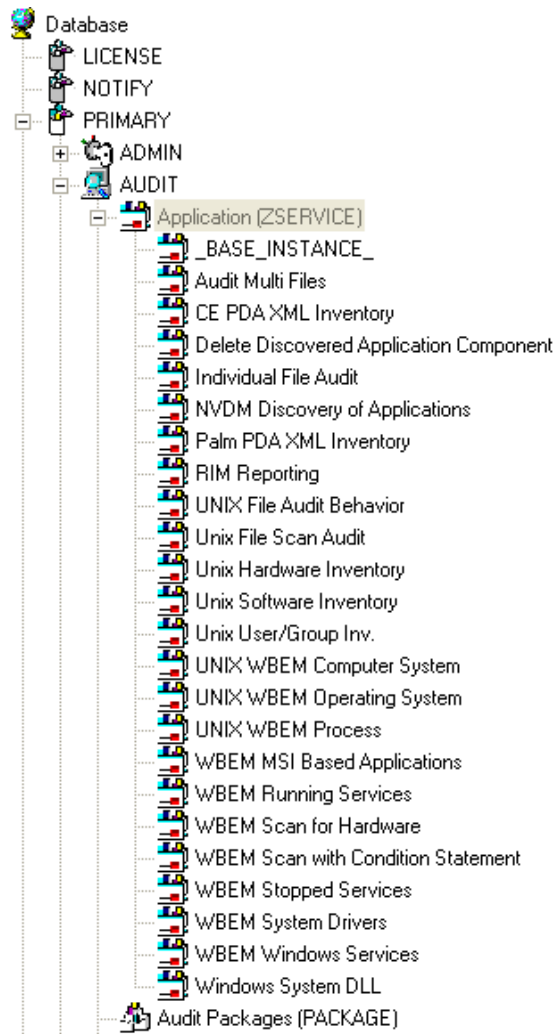
At the end of this chapter, you will:

- Know how to use the prepackaged Audit Applications (ZSERVICE).
- Know how to design your own Audit Packages (PACKAGE).

Sample Auditing

To illustrate the concepts of inventory information collection, the Inventory Manager installation contains a set of representative audit service examples. These samples are located in the PRIMARY.AUDIT.Audit Application (ZSERVICE) Class. To view these, use the Admin CSDB Editor, which is available for Windows platforms.

Figure 7 Sample Auditing Services in PRIMARY.AUDIT Domain



These sample services represent common scenarios for inventory collection and management. The best way to develop your own audit services is to study these samples.

The sample audit services are described in [Table 19](#) below.

Table 19 Sample of Auditing Services

Service	Connected to Audit Package (PACKAGE)	Description
_BASE_INSTANCE_		This service instance is the base instance for the Audit Application (ZSERVICE) Class.
Audit Multi Files	Audit to find and Capture Multiple Files	This service scans for a file name or pattern and reports that information back to the administrator.
CE PDA XML Inventory	CE PDA XML Inventory	This service scans for and reports back information on installed Windows CE PDA devices. Will only report back if a device is found.
Delete Discovered Application Component	Audit to Find and Remove Local File	This service looks for a specific file on the user's computer. If it is found, it will be deleted.
Individual File Audit	Audit to Find and Capture Local File	This service performs an NVDM scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes.
NVDM Discovery of Applications	NVDM Discovery of Applications	Used to discover software applications that are installed on an agent machine.
Palm PDA XML Inventory	Palm PDA XML Inventory	This service scans for and reports back information on installed Palm PDA devices. Will only report back if a device is found.

Service	Connected to Audit Package (PACKAGE)	Description
RIM Reporting	RIM Reporting	<p>This service performs a scan of a systems Win32 devices such as:</p> <p>BIOS, Computer System, environment, keyboard, logical disk, logical memory configuration, network adapter, operating system, pointing device, printer, processor product, serial port, service, software element, and video controller.</p> <p>Note: This is a very large scan and may take several minutes to complete.</p>
Unix File Scan Audit	UNIX File Scan Audit	This service performs a scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes on UNIX platforms.
Unix Hardware Inventory	Unix Hardware Inventory	This service scans for and reports on a user's hardware on UNIX computers.
Unix Software Inventory	Unix Software Audit	This service performs an audit to find UNIX-based software.
WBEM MSI Based Applications	WBEM Scan for Windows Installer Applications	This service performs a WBEM scan of the user's computer for components registered in the WMI database that have been installed by Microsoft Windows Installer.

Service	Connected to Audit Package (PACKAGE)	Description
WBEM Running Services	WBEM Scan for Running Services	This service scans the user's computer for system services that are running at the time of the scan.
WBEM Scan for Hardware	WBEM Scan for System Software	This service scans for and reports on a user's hardware.
WBEM Scan with Condition Statement	WBEM Scan with Condition Statement	This service performs scans based on a conditional statement set in the CONDITION attribute.
WBEM Stopped Services	WBEM Scan for STOPPED Services	This service scans the user's computer for system services that are stopped at the time of the scan.
WBEM System Drivers	WBEM Scan for Windows System Drivers	This service scans the user's computer for Win 32 system drivers.
WBEM Windows Services	WBEM Scan for Windows Services	This service scans for and reports on Windows Services.
Windows System DLL	Audit System DLL	This service scans for system DLLs and reports on them.

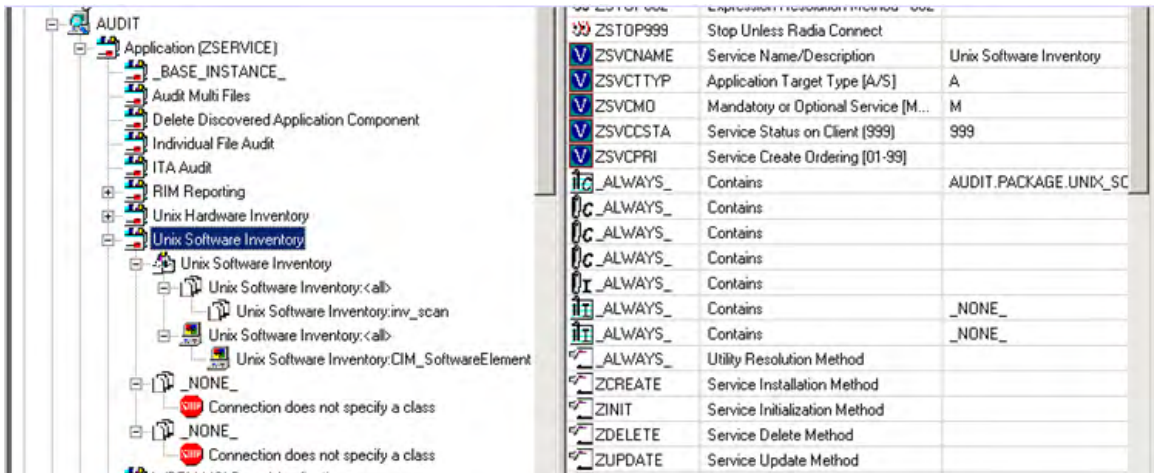
Configuring a Sample Audit

All of the examples presented can be configured for individuals, departments, work-groups, and so forth. Refer to the *Administrator Guide* for more information on manipulating the database components.

For documentation purposes, we will configure the sample audit service Unix Software Inventory. This type of audit scans for all UNIX software that is installed and managed on the agent computer. The ACTION attribute

indicates that the discovery of the file will be reported and sent to the Configuration Server for storage.

Figure 8 Unix Software sample audit in AUDIT.ZSERVICE



To configure a sample Audit package

- 1 If you have not already done so, start the Admin CSDB Editor.
- 2 Navigate to and expand the PRIMARY.AUDIT Domain.
- 3 Double-click **Application (ZSERVICE)** to expand the class.
- 4 Scroll to and expand the POLICY Domain.

For our example, we would like all users who are members of the Workgroup class to select this audit package from their Service Lists.

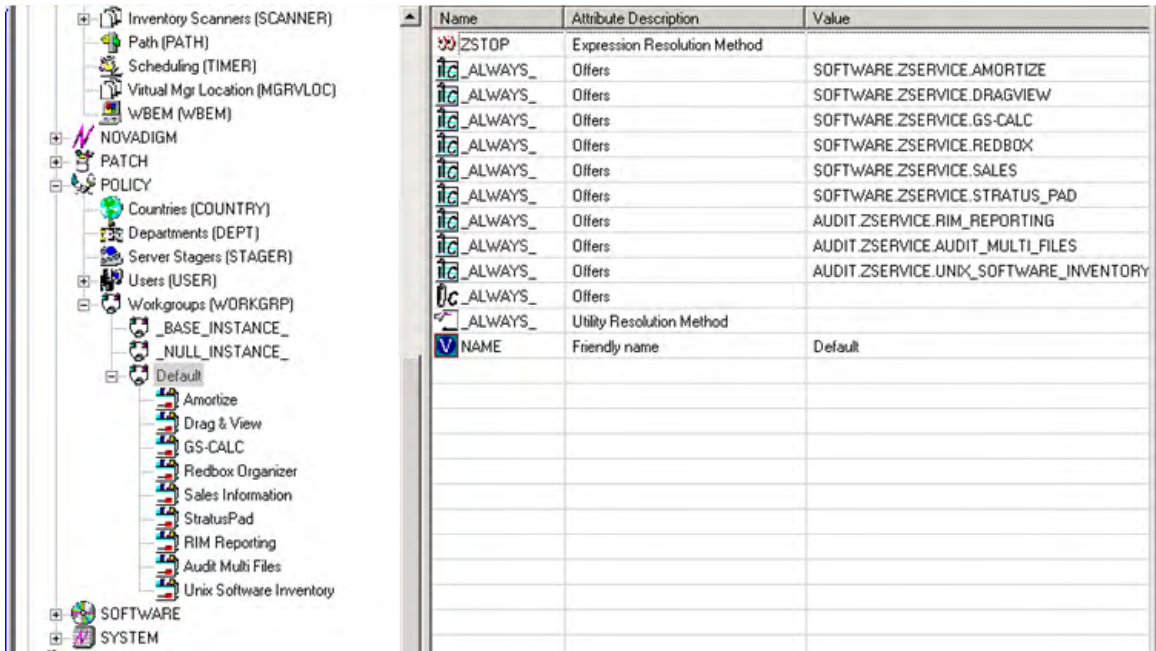
- 5 Expand the POLICY.WORKGROUPS class.
- 6 Select the **Unix Software Inventory** package from the ZSERVICE Class, drag it to the POLICY.WORKGROUPS class, and drop it on the **Default** instance. The Select Connection Attribute window opens.

- 7 Click **Copy** to add this package.

The Confirm Connection dialog box opens.

- 8 Click **Yes** to confirm the connection.

The Unix Software Inventory package is added to WORKGRP Class.



The collection of inventory information occurs on the agent computer when a user connects to the Configuration Server through the Application Manager agent when scheduled or notified to connect.

► Some scans may take several minutes to complete. This is a normal behavior of the audit scanning process.

Inventory Scan Results

Use the Admin Agent Explorer to locate the ZSERVICE instance for the Unix Software Inventory package in the LIB directory.

To locate the ZSERVICE object using the Admin Agent Explorer

- 1 Start the Admin Agent Explorer (./radobjed).
- 2 Navigate to the correct path of the Unix Software Inventory ZSERVICE instance. A sample location for the ZSERVICE object would be:

```
/opt/HP/CM/Agent/lib/SYSTEM/NVDM/SOFTWARE/ZSERVICE/UNIX_SOFTWARE_INVENTORY
```

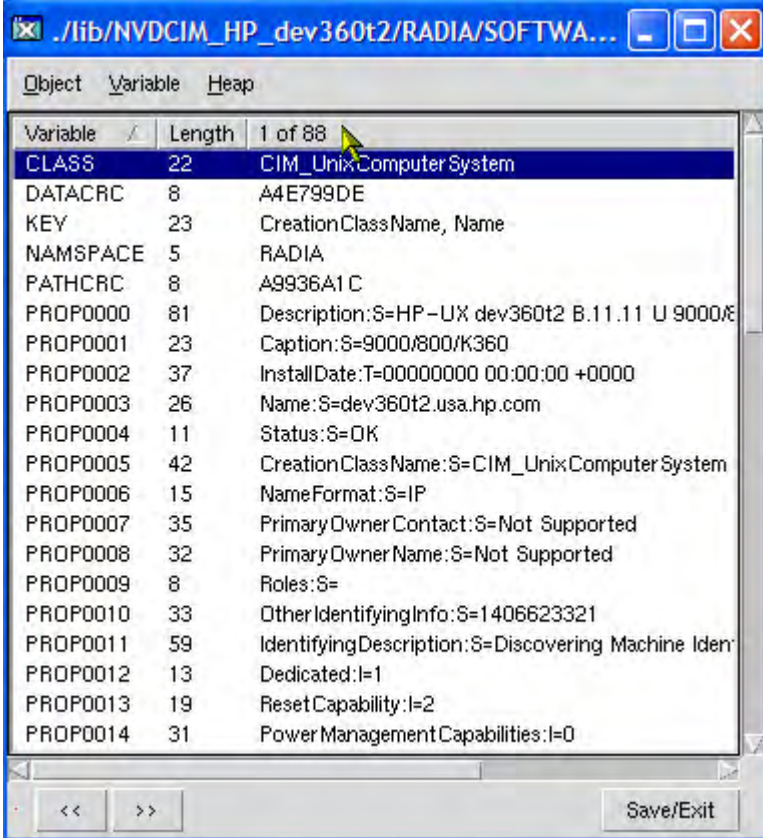
Name	Instances	Size	Modified
ZMASTER	1	8,208	Thu Dec 8 09:27:59 20
PCLSIGNO	8	8,208	Thu Dec 8 09:26:58 20
DMSYNC	1	4,624	Thu Dec 8 09:26:58 20
CONNECT	1	4,624	Thu Dec 8 09:27:57 20
APPINFO	1	6,672	Thu Dec 8 09:26:58 20
ZCONFIG	1	10 KB	Thu Dec 8 09:26:59 20
ZDSPM000	1	5,136	Thu Dec 8 09:26:59 20
TRANSFER	1	4,624	Thu Dec 8 09:26:59 20
WBEMPREV	88	246 KB	Thu Dec 8 09:27:57 20

Within the ZSERVICE, note the object WBEMPREV. This object is created and stored in the ZSERVICE of the LIB directory whenever a WBEM package is installed. The WBEMPREV object contains one heap for each file discovered during the scan. It also contains the variables from the AUDIT.WBEM instance that controlled the scan.

The AUDIT.WBEM Class instances in an audit package control the auditing for files on the agent computer.

- The agent scans the client's computer file system based upon the values contained in the AUDIT.WBEM Class instance in the audit package. It constructs an object called WBEMCURR.
- The WBEMCURR object contains one heap per instance of each WBEM Class discovered during the current scan.
- The agent compares the scan results from the current scan (the scan done during the current agent connect stored in the WBEMCURR object) with the scan results from a previous scan (the scan done during a previous agent connect process stored in the WBEMPREV object). It will construct the WBEMAUDT object that is then sent to the Configuration Server.
- The agent then deletes the WBEMAUDT object and will rename the WBEMCURR object to WBEMPREV.

Figure 9 WBEMPREV heaps in Admin Agent Explorer



Variable	Length	1 of 88
CLASS	22	CIM_UnixComputerSystem
DATA_CRC	8	A4E799DE
KEY	23	CreationClassName, Name
NAMESPACE	5	RADIA
PATH_CRC	8	A9936A1C
PROP0000	81	Description:S=HP-UX dev360t2 B.11.11 U 9000/800/K360
PROP0001	23	Caption:S=9000/800/K360
PROP0002	37	InstallDate:T=00000000 00:00:00 +0000
PROP0003	26	Name:S=dev360t2.usa.hp.com
PROP0004	11	Status:S=OK
PROP0005	42	CreationClassName:S=CIM_UnixComputerSystem
PROP0006	15	NameFormat:S=IP
PROP0007	35	PrimaryOwnerContact:S=Not Supported
PROP0008	32	PrimaryOwnerName:S=Not Supported
PROP0009	8	Roles:S=
PROP0010	33	OtherIdentifyingInfo:S=1406623321
PROP0011	59	IdentifyingDescription:S=Discovering Machine Ident
PROP0012	13	Dedicated:I=1
PROP0013	19	ResetCapability:I=2
PROP0014	31	PowerManagementCapabilities:I=0

For our particular example, there were 318 instances for the WBEMPREV object located on the subscriber's computer.

7 Creating Audit Packages

At the end of this chapter, you will:

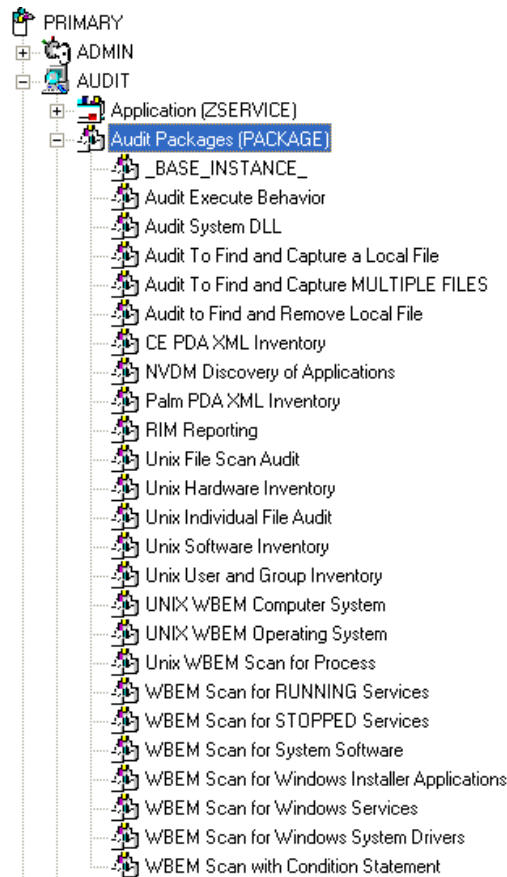
- Have created a new file audit package.
- Have created a new ZSERVICE for your package.

Audit Packages (PACKAGE) Class

Once you are comfortable auditing using the sample packages provided by HP, take the next step in designing your own audit packages.

By expanding the Audit Packages (PACKAGE) Class, you will refer to the audit package instances.

Figure 10 Audit Packages (PACKAGE) Class



A complete audit service consists of several connected instances in the AUDIT Domain. The audit package instance is a container that "owns" the instances connected to it.

For example, open the AUDIT.ZSERVICE Class and double-click the **UNIX Hardware Inventory** instance.

Figure 11 Unix Hardware Inventory instance



In the example, the UNIX Hardware Inventory ZSERVICE instance "owns" the UNIX Hardware Inventory instance. The fact that a package instance owns a component class instance means that all of the instances are managed as a package unit. If the package instance is deleted, all of its owned class instances are automatically deleted as well.

▶ Sound database management practices dictate that the component class instances owned by a package are not connected to any other package instance.

The audit service instance must also contain a connection to an instance of the RIMOPTS Class. Connecting an instance of the RIMOPTS Class to an audit service instance causes the expressed behavior to be performed. Specified behaviors are listed in the following table.

Table 20 Inventory Options (RIMOPTS) Class

Instance	Description
Default	Contains the base instance attributes for the RIMOPTS Class. <ul style="list-style-type: none"> • Collect attribute is set to Diff. • Runexec attribute is set to IU. • ZSVCTYPE attribute is set to I.
Differenced Audit on Install and Update	When connected to an audit service will difference the audited information on installation and when the audited target is updated. <ul style="list-style-type: none"> • Collect attribute is set to Diff. • Runexec attribute is set to IU. • ZSVCTYPE attribute is set to I.

Instance	Description
Differenced Audit on Install, Verify, and Update	<p>When connected to an audit service, will difference the audited information in initial installation, on subsequent connects, and when updated.</p> <ul style="list-style-type: none"> • Collect attribute is set to Diff. • Runexec attribute is set to IVU. • ZSVCTYPE attribute is set to I.
Full Audit on Install and Update	<p>When connected to an audit service, will difference the audited information on installation and update.</p> <ul style="list-style-type: none"> • Collect attribute is set to Full. • Runexec attribute is set to IU. • ZSVCTYPE attribute is set to I.
Full Audit on Install, Verify and Update	<p>When connected to an audit service, will</p> <ul style="list-style-type: none"> • Collect attribute is set to Full. • Runexec attribute is set to IVU. • ZSVCTYPE attribute is set to I.

Using Admin CSDB Editor to Create and Maintain Audit Services

We will use the Admin CSDB Editor to walk through the construction of a file audit. The inventory information to collect, and the action to take with that collected information, is specified in an instance of the AUDIT Domain's Audit Packages (PACKAGE) Class.



The Admin CSDB Editor is available for Windows platforms. For more information, refer to the *Administrator Guide*.

Prior to beginning the creation of the package, you should ask yourself the following questions:

- What am I auditing for? Will it be a hardware audit, a file audit, or a WBEM object audit?

- Will I be deploying to all users, or a select few?
- Will I want this to be connected to a timer for scheduled deployment?

By viewing and deploying the sample audits provided by HP, you will be able to create and use your own auditing packages.

To create a new Audit package

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

The Security Information dialog box opens.



The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation. Check with your HPCA security administrator to obtain your own User ID and Password, if necessary.

- 2 If necessary, type a User ID and Password, and then click **OK**. The Admin CSDB Editor window opens.
- 3 Double-click **PRIMARY**.
- 4 Expand the AUDIT Domain.
- 5 Double-click the **Audit Packages (PACKAGE) Class**.
As an example, we will create a new auditing package called ITA Audit Package. This package will scan a user's computer, capture logical disk information, and return the results to the administrator.
- 6 Right-click the **Audit Packages (PACKAGE) Class** and select New Instance from the shortcut menu. The Create Instance dialog box opens.
- 7 In the upper text box, type a new display name for the package instance. This is the friendly name that will appear in the tree view.
- 8 In the lower text box, type a name for the Create a new Audit Packages (PACKAGE) instance named. This is the name that appears in the title bar of the list view (right side) of the Admin CSDB Editor window when the instance is selected and opened in the tree view.
- 9 Click **OK** to continue.

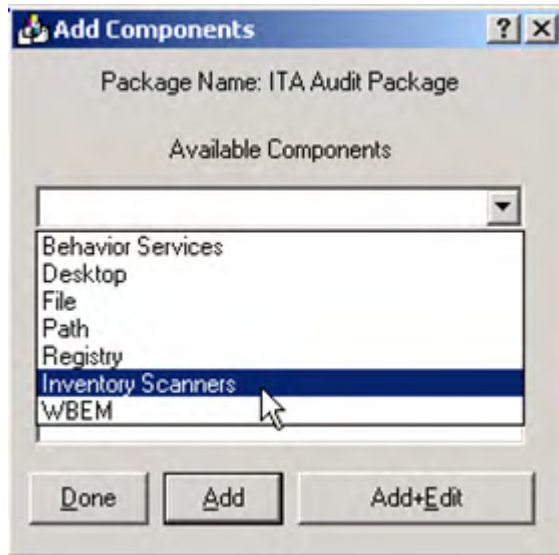
The new Audit package is added to the AUDIT.PACKAGE Class.



Once the Audit package is created, you will need to add its components.

To add a component to an Audit package

- 1 Right-click the new Audit package.
- 2 Select **Add Components** from the shortcut menu. The Add Components dialog box opens.
- 3 Click the Available Components down arrow.



- 4 From the list that opens, select **Inventory Scanners**.
- 5 In the New Component Name text box, type the name of the new component.
- 6 Click **Add+Edit**. The component is added to the package and the Editing Instance dialog box opens.

In the Editing Instance dialog box you can edit the instances that will be used in your audit.

- 7 Scroll down to the PARMs attribute and select it.
- 8 In the Parameters text box type **nvdcm**. This is the name of the Tcl script that will be executed by the client to initiate the inventory scan. The nvdcm Tcl script is included with the UNIX Inventory material.



A connection to the Scanner class may be used to run any custom client inventory method.

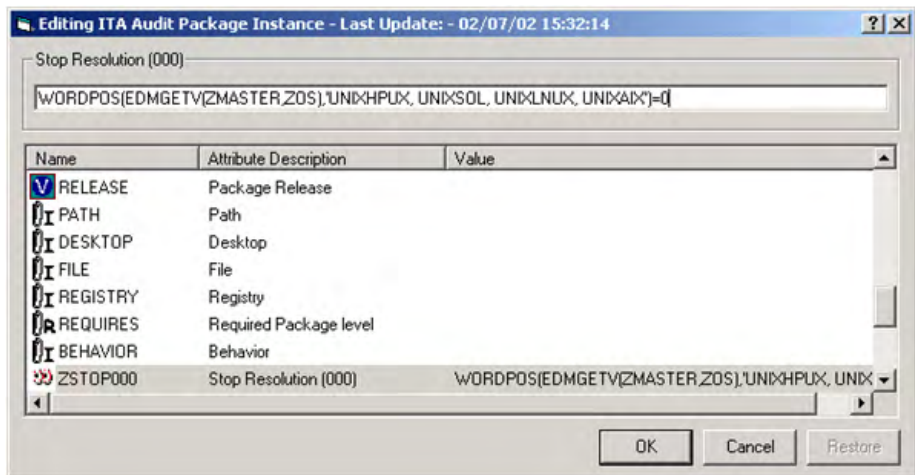
- 9 Click **OK** when you are done with your edit. You return to the Add Components dialog box.
- 10 Next, add a WBEM Class component to the package. You will need to add a WBEM Class component for each inventory shell script you execute.
- 11 From the Available Components drop-down list, select WBEM.
- 12 In the New Component Name text box, type the name of the WBEM component.

- 13 Click **Add+Edit**. The component has been added to the package and the Editing CIM_LogicalDisk Instance dialog box opens.
- 14 Select the **CLASS** attribute, and in the Class text box type **CIM_LogicalDisk**. This is the name of the file that will be used to execute the inventory collection. CLASS is the only attribute used by the client Inventory Harness.
- 15 When finished, click **OK**.
- 16 Click **Done** in the Add Components dialog box.

Now edit the package class instance ZSTOP expression to reflect the supported UNIX platforms. The default ZSTOP expression is configured for Windows platforms.

To update the ZSTOP expression

- 1 In the tree view of the Admin CSDB Editor, double-click the new audit package name, **ITA Audit Package**.
- 2 In the list view of the Admin CSDB Editor, double-click the **ZSTOP** expression.
- 3 Replace the supported Windows platform names with the appropriate UNIX platforms.



- 4 Click **OK**.

To create a ZSERVICE instance

Next, you will need to create a ZSERVICE instance to contain the package.



While working within the AUDIT Domain, note that the New Application Wizard is *not* available to connect a package to a service. You need to either copy an existing instance or create a new one.

- 1 In the Admin CSDB Editor, expand the AUDIT.ZSERVICE Class.
- 2 Right-click **Audit Application (ZSERVICE)** and a shortcut menu opens.
- 3 Select **New Instance** from the shortcut menu.
- 4 Type a display name and an instance name.
- 5 Click **OK**. The ZSERVICE is added to the AUDIT.ZSERVICE Class.

Use the Admin CSDB Editor to connect the new ZSERVICE instance to the Audit Package.

Now, add `_NONE_` to the RIMOPTS and BEHAVIOR connections. These are default connections from the base instance and are only applicable to Windows clients.

- 6 Double click the ZSERVICE instance.
- 7 Double-click the two class connections and change their values to `_NONE_`.
- 8 Click **OK**.

Creating UNIX File Audit Methods

Unix File Audit methods are run for reporting purposes. The AUDIT Classes FILESCAN and FILTER are used when creating Unix File Audit methods. Creating a new Unix File Audit method is similar to creating a new package for inventory scanning, as seen in the previous section.

To create a new Unix File Audit method package

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

The Security Information dialog box opens.



The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation. Check with your HPCA security administrator to obtain your own User ID and Password, if necessary.

- 9 If necessary, type a User ID and Password, and then click **OK**. The Admin CSDB Editor window opens.
- 10 Double-click **PRIMARY**.
- 11 Expand the **AUDIT Domain**.
- 12 Double-click **Audit Packages (PACKAGE) Class**.
As an example, we will create a new auditing package called **Unix File Audit**. This package will scan a user's computer.
- 13 Right-click the **Audit Packages (PACKAGE) Class**.
- 14 Select **New Instance** from the menu.
- 15 Type a new display name for the package instance. This is the friendly name that will appear in the tree view.
- 16 Type a name for the Create a new Audit Packages (PACKAGE) instance named. This name appears in the title bar of the list view of the Admin CSDB Editor window when the instance is selected and opened in the tree view.
- 17 Click **OK** to continue. The new Audit Package is added to the **AUDIT.PACKAGE Class**.
- 18 After you create the Audit package, add the components for the Unix File Audit method.

To add a component to an audit package

- 1 Right-click on the new Audit package.
- 2 Select **Add Components** from the context menu. The Add Components dialog box opens.
- 3 Click the **Available Components** down arrow. Select **File Scanner** from the list.
- 4 In the New Component Name text box, type the name of the component.
- 5 Click **Add+Edit**. This adds the component to the package and opens the Editing Instance dialog box.
Use the Editing Instance dialog box to edit the instances used in your file scan.
- 6 Click **OK** when you are finished editing your instance.
- 7 Now add a File Scanner Filters component.

- 8 From the Available Components drop-down list, select **File Scanner Filters**.
- 9 In the New Component Name text box, type **File Scanner Filters**.
- 10 Click **Add+Edit** to add the component to the package and open the Editing Instance dialog box.
- 11 Click **OK** when you are finished editing the instance.
- 12 Click **Done** in the Add Components dialog box.
- 13 Now create a ZSERVICE instance and connect the package. Make sure to add `_NONE_` to the two ALWAYS connections in the ZSERVICE instance. See [To create a ZSERVICE instance](#) on page 92 for instructions on creating a ZSERVICE and removing the required ALWAYS connections.

8 Configuring Timers for Audit Collection

At the end of this chapter, you will:

- Have created an Audit `TIMER` instance for an audit package.
- Have created an Audit `TIMER ZSERVICE` for an audit package.

The Scheduling (TIMER) Class

The Scheduling (TIMER) Class enables the administrator to set a timer on the agent computer and will cause one or more audit services to be processed whenever the timer expires. The administrator can use this method to process mandatory audit services automatically according to a predetermined schedule.

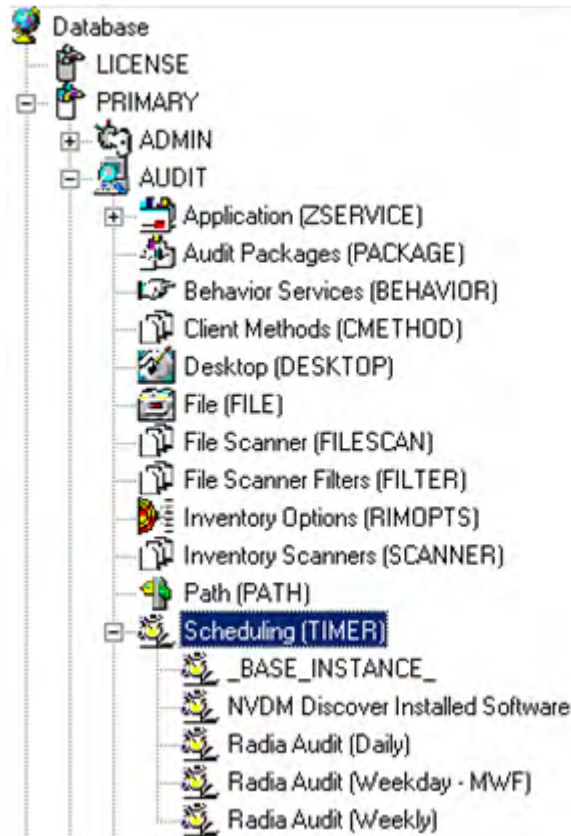
▶ As distributed by HP, the SOFTWARE Domain also contains a Scheduling (TIMER) Class. Timers can be specified in instances of either Scheduling (TIMER) Class and can be connected to an Application (ZSERVICE) Class instance in either the SOFTWARE or AUDIT Domains interchangeably.

Housed within the AUDIT.Scheduling (TIMER) Class are three sample Timer packages:

- **Daily**
which will deploy a ZSERVICE everyday at the time specified.
- **Weekday**
which will deploy a ZSERVICE on Mondays, Wednesdays, and Fridays at a specified time.
- **Weekly**
which will deploy a ZSERVICE every seven days at a specified time.

These sample packages can be copied, changing the time parameters to suit your needs. Refer to the *Administrator Guide* for information on copying an instance. Or, you can create a new timer instance by following the procedure [To create a new timer in the AUDIT Domain](#) beginning on page 103.

Figure 12 AUDIT Scheduling (TIMER) Class



Timers can be set to expire periodically (hourly, daily, weekly, monthly, or at defined intervals), on a specific date, or at a specific time. Each agent is installed with the Scheduler service. This service contains an executable timer component that executes any program on the end-user desktop when a timer expires.

Typically, the Scheduler service lies dormant in the background, and wakes up once per minute to see if a timer has expired. When a timer expires, the command line associated with the expired timer is executed. Normally, this command line invokes a connection to the Configuration Server to deploy or maintain a service.

The following table explains the Scheduling (TIMER) Class attributes:

Table 21 Scheduling (TIMER) Class

Attribute	Description
ZOBJPRI	Sets the priority for deployment of ZTIMEQ object, which is deployed relative to the other elements being deployed during the agent connect. Elements with a priority number less than the value of ZOBJPRI are deployed <i>before</i> the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.
ZSTOP	Used to assign timer conditions. Indicate true to cause resolution of the instance to be skipped. The timer is not deployed for end users. Leave blank for the instance to be accepted, and resolution will continue.
ZSCHMODE	Specifies the timer owner. It is recommended that you accept the default configuration of USER.
ZSCHDEF	Indicates when timer expires. The syntax varies depending on the frequency of expiration, which can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.
ZSCHTYPE	<p><i>Used only when ZSCHFREQ = PERIODIC.</i></p> <p>Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4 AM (non-peak) scheduled event that is sent to the agent computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning.</p> <p>Example 1:</p> <p>Suppose you create and deploy a timer with the ZSCHDEF = DAILY(&ZSYSDATE,04:00:00)</p> <p>If ZSCHTYPE = IMMEDIATE and it is:</p> <ul style="list-style-type: none">• Before 04:00:00, the command in the instance will be executed the same day at 04:00:00• After 04:00:00, the command in the instance will be executed immediately <p>If ZSCHTYPE = DEFERRED and it is:</p> <ul style="list-style-type: none">• Before 04:00:00, the command in the instance will be executed the <i>next</i> day at 04:00:00• After 04:00:00, the command in the instance will be

Attribute	Description
	<p>executed the <i>next</i> day at 04:00:00</p> <p>Example 2:</p> <p>Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,04:00:00)</p> <p>If ZSCHTYPE = IMMEDIATE and it is:</p> <ul style="list-style-type: none"> • Not Friday or Friday and before 04:00:00, the command in the instance will be executed on Friday at 04:00:00 • Friday and after 04:00:00, the command in the instance will be executed immediately <p>If ZSCHTYPE = DEFERRED and it is:</p> <ul style="list-style-type: none"> • Not Friday or Friday and before 04:00:00, the command in the instance will be executed a week later on Friday at 04:00:00 • Friday and after 04:00:00, the command in the instance will be executed a week later on Friday at 04:00:00
ZSCHFREQ	<p>Indicates how often the timer should expire according to the frequency specified in the ZSCHDEF attribute.</p> <ul style="list-style-type: none"> • Once for a one-time expiration. • Periodic for a repeated expiration. • Random for random intervals.
ZRSCCMDL	<p>Indicates the command line that is executed on the subscriber's computer when the timer expires.</p>
ZSVCOID	<p>Specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.</p>
ALWAYS	<p>Stores the connections to other instances.</p>
NAME	<p>The friendly name for this instance.</p>
APPSVC	<p>The Application Name.</p>
REQUEST	<p>The Application Request.</p>
DOMAIN	<p>The server's domain name.</p>
IPADDR	<p>The server's IP address/name.</p>
SOCKET	<p>The server's socket number.</p>

Attribute	Description
MGRNAME	The server's name.
ZCREATE	The Scheduler CREATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZVERIFY	The Scheduler VERIFY method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZUPDATE	The Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZDELETE	The Scheduler DELETE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZNOPING	Controls the automatic sensing of a network connection between the agent computer and the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. Use this attribute when there is a possibility that the client will not be able to connect with the Configuration Server, such as when the client is a mobile user. Note: In order to use this attribute, you must add it to the TIMER Class template.

Creating a Timer Instance

This section covers how to create and configure a timer and connect it to the service that you want to deploy. Prior to creating and configuring a timer, consider the following:

- What time of day should the timer expire?
- How often do you want the timer to expire?
- Does the timer need to expire more than once?
- What should happen when the timer expires?

To create a timer, use the Admin CSDB Editor to create a Scheduling (TIMER) instance in the AUDIT Domain.

▶ As distributed by HP, the SOFTWARE Domain also contains a Scheduling (TIMER) Class. Timers can be specified in instances of either Scheduling (TIMER) Class and can be connected to an Application (ZSERVICE) Class instance in either the SOFTWARE or AUDIT Domains interchangeably.

For the purposes of documentation, the timer created will be created from within the AUDIT Domain.

For more information concerning the Schedule (TIMER) Class, refer to the *Deploying Services* chapter of the *Application Manager Guide*.

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

To create a new timer in the AUDIT Domain

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

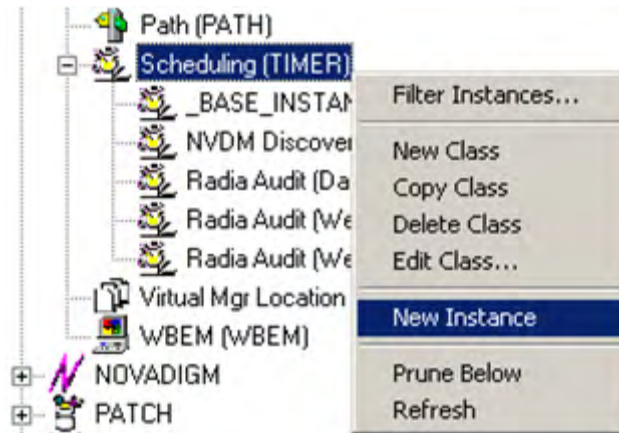
The Admin CSDB Editor Security Information dialog box opens.

▶ The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation. Check with your HPCA security administrator to obtain your own User ID and Password, if necessary.

- 2 If necessary, type a User ID and Password, and then click **OK**. The Admin CSDB Editor window opens.
- 3 Double-click **PRIMARY**.

- 4 Expand the **AUDIT Domain**.
- 5 Right-click **Scheduling (TIMER)**.

A shortcut menu opens.



- 6 Select **New Instance**. The Create Instance dialog box opens.
- 7 Type a name for the new timer instance.
- 8 Click **OK**. The new timer instance appears in the Scheduling (TIMER) Class.

Specifying Timer Settings

Whether you copied an existing timer or you created a new Timer instance, you will need to review and/or customize your timer settings.

- ▶ Refer to the *Deploying Services* chapter in the *Application Manager Guide* for more Schedule (TIMER) Class information.

Specifying ZSCHDEF

Use ZSCHDEF to indicate when the timer should expire. The syntax varies depending upon the expiration frequency. When configuring ZSCHDEF, the variable is set in the following form:

```
freq(date,time[,limit_time][count])
```

- The value of *freq* can be:

DAILY, WEEKLY, WEEKDAY, HOURLY, INTERVAL, NUMDAYS

- If the value of *freq* is DAILY, WEEKLY, HOURLY, INTERVAL, or NUMDAYS, the date is then specified in the following form:

YYYY/MM/DD

- If the value of *freq* is WEEKDAY, the date is then specified as the name of a day of the week in all uppercase letters. This would be one of the following:

MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

- The value for *time* or *limit_time* is optional. It is specified in the following form:
HH:MM:SS
- The value for *count* is optional. It is specified as an integer.
- The timer expiration can also be configured on the value of ZSCHFREQ. Use [Table 22](#) below to help you determine the appropriate syntax.

Table 22 Syntax of ZSCHDEF Variables

Type	Syntax	Timer Expires
DAILY	DAILY(&ZSYSDATE,24:00:00)	Daily at midnight by the system's date.
WEEKLY	WEEKLY(&ZSYSDATE,01:00:00)	Every 7 days at 1:00 AM.

Type	Syntax	Timer Expires
WEEKDAY	WEEKDAY(MONDAY,01:00:00)	Every <i>Name of Weekday</i> * starting on MONDAY at 1:00 AM. The weekday must be specified in uppercase.
HOURLY	HOURLY(&ZSYSDATE,08:41:00)	Hourly starting at 8:41 AM on the systems date.
INTERVAL	INTERVAL(&ZSYSDATE,08:41:00,,30)	Every 30 minutes starting at 8:41 AM based on system's date.
NUMDAYS	NUMDAYS(20000803,08:00:00,,14)	Every 14 days starting on August 3, 2000 at 8:00 AM.

* *Name of Weekday* is the name of a specific weekday in uppercase letters, e.g. MONDAY.

Specifying ZSCHTYPE

The ZSCHTYPE controls how the timer handles the scheduled event when the client receives the initial TIMER definition for a service. There are two valid controls:

- **IMMEDIATE**
will execute the command specified in the ZRSCCMDL immediately if the date and time indicated in ZSCHDEF has passed when the ZTIMEQ object is initially created.
- **DEFERRED**
will defer the execution if the date and time defined in the ZSCHDEF has passed and will wait until the next occurrence to execute. This is the recommended setting.

If the time and date indicated in ZSCHDEF has not passed when the ZTIMEQ object is deployed, this setting has no effect.

Specifying ZSCHFREQ

Use ZSCHFREQ to specify whether the timer should expire once (ONCE) or repeatedly (PERIODIC) according to the frequency specified in ZSCHDEF.

Specifying ZRSCCMDL

Use ZRSCCMDL to execute a command on the subscriber's computer when the timer expires.

Use the following command line to run the audit service when the scheduled time occurs:

```
radskman,cat=y,uid=&(ZMASTER.ZUSERID),startdir=&(ZMASTER
.LOCALUID),mname=&(ZMASTER.ZMGRNAME),dname=&(ZMASTER.ZDOMNAME,
sname=&(ZSERVICE.ZOBJNAME)
```



The parameters indicated in the `radskman` command may differ depending upon customer specific implementations.

Specifying ZNOPING

The ZNOPING attribute controls automatic sensing of a network connection between the agent computer and the Configuration Server. Use this attribute when there is a possibility that the client will not be able to connect with the Configuration Server, such as when the client is a mobile user.

- If the ZNOPING attribute is not in the ZTIMEQ object, or if ZNOPING is not equal to N, the Scheduler service does not ping the Configuration Server.
- If ZNOPING = N, the Scheduler service will ping the Configuration Server.
 - If the Configuration Server is pinged successfully, the command in ZRSCCMDL is executed. The PENDING attribute in the client's ZTIMEQ object is then set to N. This will 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. The PENDING attribute value remains set to Y. The next time the Scheduler service expires, it should ping the Configuration Server again.

Connecting the Timer to a Service

Once you have created your timer, you must connect it to a service. Each subscriber that receives the ZSERVICE to which the timer is connected, will

receive the timer information in the ZTIMEQ object the next time the agent connects to the Configuration Server.

Use the Admin CSDB Editor to connect the ITA Audit Timer to the ITA Audit ZSERVICE created earlier in this document.

Then connect the AUDIT.ZSERVICE .ITA Audit to a user or group of users within the POLICY Domain.

Audit Execution Configuration

By default, when an Audit service is installed on an end user's computer, it executes immediately and reports to the Configuration Server. This can be time consuming, especially if the audit service type is WBEM or File Scan. The audit service definition may also be installed at a time when an audit scan is not desirable. For example, when an end user visits the Application Self-service Manager and mandatory applications are processed as defined in the embed tag `enterprisemanagement=auto`.

The easiest way to approach this issue is to manipulate how and when the audit actually executes. This can be accomplished by:

- Customizing the Inventory Options (RIMOPTS) attribute.
and
- Updating the embed tags in the html file for the Application Self-service Manager.

The following describes the steps necessary to customize RIMOPTS and update the embed tag to prevent audit execution during mandatory application processing.

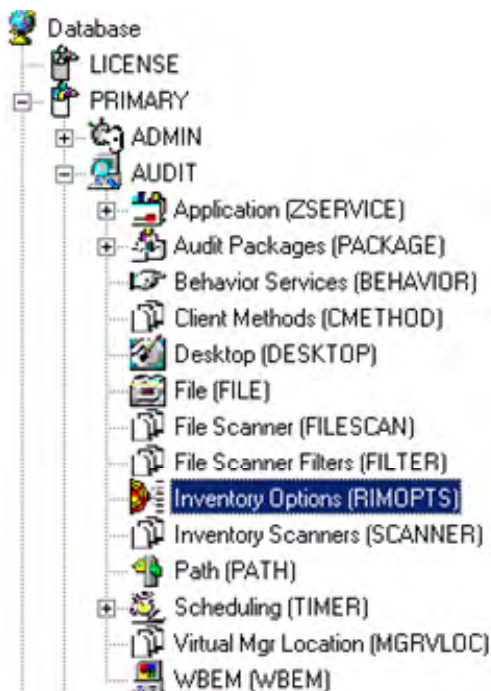
To customize the RIMOPTS instance

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

The Security Information dialog box opens.

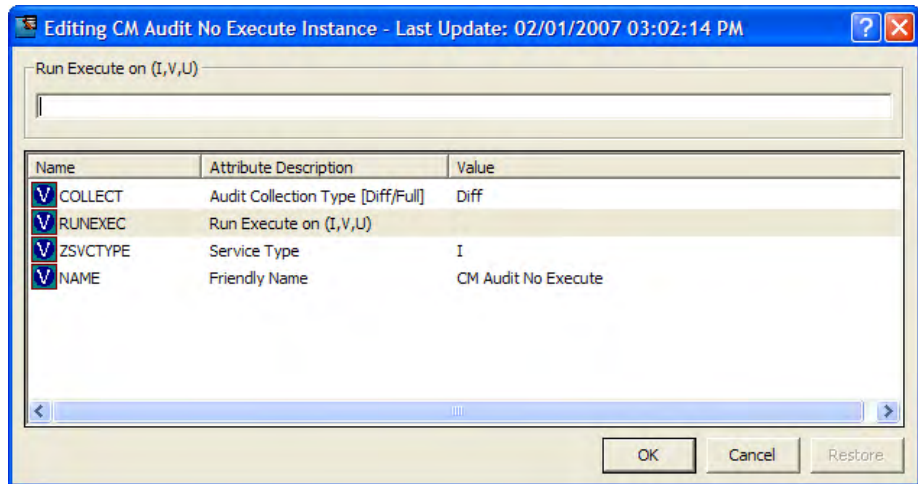
▶ The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation. Check with your HPCA security administrator to obtain your own User ID and Password, if necessary.

- 2 If necessary, type a User ID and Password, and then click **OK**. The System Explorer window opens.
- 3 Expand the **PRIMARY File** and the **AUDIT Domain**.



- 4 Create a new instance in the Inventory Options (RIMOPTS) Class called CM_AUDIT_NO_EXECUTE, and click **OK**.
- 5 Expand the Inventory Options (RIMOPTS) Class and double-click the **CM Audit No Execute** instance.

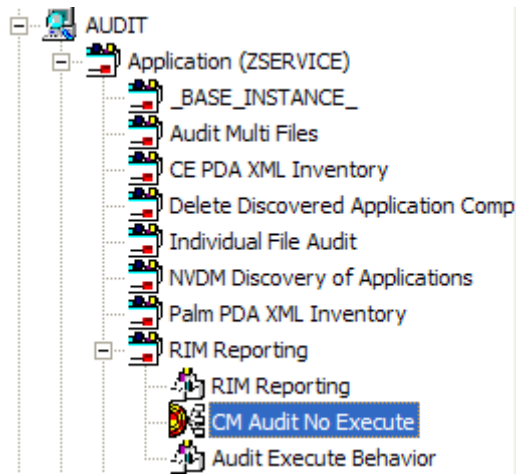
- 6 Double-click the **RUNEXEC** attribute in the list view to edit it. Remove any attribute information. This will ensure that the audit service will not run during the installation, verification, or update function.



Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the RIM_REPORTING service.

- 7 Right-click the **RIM_REPORTING** Service in the AUDIT Class.
- 8 Select **Edit Instance**.
- 9 Locate the **_ALWAYS_ Contains** attribute with the value of `AUDIT.RIMOPTS.DIFF_INSTALL_UPDATE` and change it to a value of **AUDIT.RIMOPTS.CM_AUDIT_NO_EXECUTE**.
- 10 Next, to define the audit service as Mandatory, locate the **ZSVCMO** field and set it to M. This will cause the initial TIMER definition associated with the audit service to be created on the agent.

The **CM Audit No Execute** instance is now connected to the RIM Reporting service, as shown in the following screen capture.



9 Viewing Inventory from the Reporting Server

At the end of this chapter, you will:

- Know how to access and use the Reporting Server to view the hardware, software, and operational information obtained from agent computers.
- Be able to navigate through the information collected by clicking hyperlinks embedded within any table.

Accessing the Reporting Server

To access the Reporting Server

- Open a Web browser and type the following address:

```
http://<hostname>/reportingserver
```

Where *<hostname>* is the host name of the Apache web server on which the Reporting Server was installed and where *reportingserver* is the alias assigned to Reporting Server during its installation.



Reporting is optimized for display screen area setting 1024 x 768 or greater.

Viewing Audit Information Using the Reporting Server

The Reporting Server provides web-based reports for Inventory Manager. For installation and configuration instructions for the Reporting Server, refer to the *Reporting Server Guide*. The Reporting Server installation media is included with the HPCA Infrastructure media.



Inventory reports may need to be enabled. This is done using the Reporting Server configuration file (*setup.tcl*). Refer to the *Reporting Server Guide* for more details.

Reporting Views for Inventory Reports

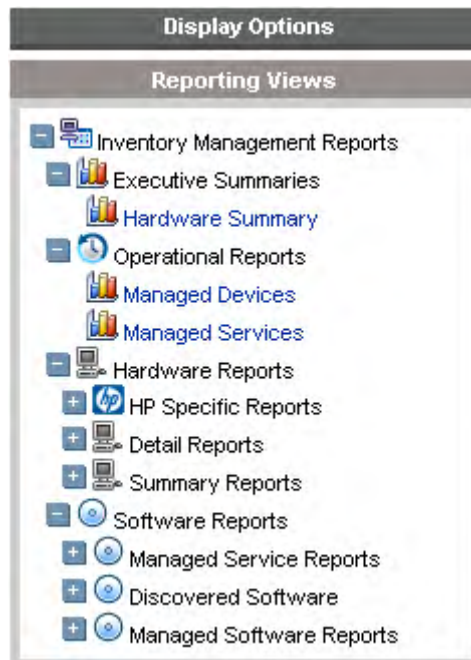
To view the reports, first access your Reporting Server. Then, under Reporting Views, click **Inventory Management Reports** to expand the list of reports.

There are different types of inventory reports:

- Executive Summaries
- Operational Reports
- Hardware Reports

- Software Reports.

Figure 13 Inventory Management Reports



The following tables list the available Hardware and Software Reporting Views.

Table 23 Hardware Reporting Views

Reporting View Types	Reporting Views
HP Specific Reports	HP BIOS Settings HP Hardware Alerts HP Hardware Alerts (Boot Events)
Detail Reports	Hardware Summary Managed Devices Devices by Vendor/Model Devices by Serial # Device by Baseboard ID Device by Logical Disks Battery Information

Reporting View Types	Reporting Views
	SMBIOS Information
Summary Reports	Count by Summary Count by CPU Count by Memory Count by Operating System

Table 24 Software Reporting Views

Reporting View Types	Reporting Views
Managed Service Reports	Service Summary Service Details
Discovered Software	Vendor Reports <ul style="list-style-type: none"> Discovered Software by Vendor Product Reports <ul style="list-style-type: none"> Discovered Software by Product Discovered Software by Version Application Reports <ul style="list-style-type: none"> Discovered Software by Application Discovered Software by Application Version
Managed Software Reports	Vendor Reports <ul style="list-style-type: none"> Managed Software by Vendor Product Reports <ul style="list-style-type: none"> Managed Software by Product Managed Software by Product Version Application Reports <ul style="list-style-type: none"> Managed Software by Application Managed Software by Application Version

Filtering Inventory Reports with Reporting Server

Reporting Server provides extensive filtering capabilities. To access the filters, expand Inventory Manager Related in the Search Controls section of the Reporting Server page.

Filter types include:

- Operational Related
- Hardware Related
- Software Related
- OS Related

Figure 14 Inventory Management Related Data Filters



Expand each individual Inventory Management Related Data Filter to refer to the available filters you can apply to the current Reporting View.

For more information on creating filters and using the Reporting Server in general, refer to the *Reporting Server Guide*.

Index

A

- ACCESSDT attribute, 60
- accessing Reporting Server, 114
- ACCESSTM attribute, 60
- ACTION attribute, 55, 62, 67, 79
- action flags, 47
- ACTION instance, 47
- agent
 - installing
 - local computer, 28
- agent computer, definition, 13
- ALWAYS attribute, 101
- APPSVC attribute, 101
- Audit Application Class, 45
- AUDIT Domain, 45
- Audit Multi Files, 77
- Audit package
 - adding components, 90
 - creating, 89
- Audit Packages Class, 45, 86, 88
- AUDIT.FILE class
 - ACTION, 55
 - GROUP, 56
 - INTERNAL, 57
 - LANGUAGE, 57
 - LOCATION, 58
 - NAME, 58
 - ORGNAME, 57
 - OUTPUT, 56
 - PRODUCT, 57
 - PRODVERS, 57
 - SCANFOR, 55
 - TYPE, 56
 - VENDOR, 57
 - VERSION, 57
 - ZCRCINFO, 58
 - ZINIT, 58
 - ZRSCCRC, 58
 - ZRSCMFIL, 57
 - ZRSCMMEM, 57
 - ZRSCOBJN, 58
 - ZRSCPADM, 58
 - ZRSCSRC, 58
 - ZRSCSTYP, 57
 - ZRSCVLOC, 57
 - ZVERINFO, 56
- AUDIT.FILE class instance, 54
- AUDIT.FILE instance, 56
- AUDIT.FILESCAN class instances, 61
- AUDIT.WBEM Class, 62
- AUDIT.WBEM instance, 65
 - ACTION, 62, 67
 - CLASS, 63, 67
 - CNDITION, 63, 67
 - NAME, 63, 68
 - NAMESPACE, 63, 67
 - OUTPUT, 63, 67
 - PROPERTY, 63, 67
 - TYPE, 63, 67
- Auditing Services
 - Audit Multi Files, 77
 - CE PDA XML Inventory, 77
 - default, 77
 - Delete Discovered Application Component, 77
 - Individual File Audit, 77
 - NVDM Discovery of Applications, 77
 - Palm PDA XML Inventory, 77
 - RIM Reporting, 78
 - Unix File Scan Audit, 78
 - Unix Hardware Inventory, 78
 - Unix Software Inventory Audit, 78
 - WBEM MSI Based Applications, 78

- WBEM Running Services, 79
- WBEM Scan For Hardware, 79
- WBEM Scan with Condition Statement, 79
- WBEM Stopped Services, 79
- WBEM System Drivers, 79
- WBEM Windows Services, 79
- Windows System DLL, 79

auditing types, 54

B

BEHAVIOR, 93

BEHAVIOR Class, 45

Behavior Services Class, 45

C

CE PDA XML Inventory, 77

-cfg, 36

CIM. See Common Information Model

CIM_CDROMDrive, 53

CIM_Controller, 52

CIM_Dependency, 52

CIM_Directory, 53

CIM_DiskDrive, 53

CIM_DVDDrive, 53

CIM_EthernetAdapter, 53

CIM_Export, 54

CIM_IDEController, 53

CIM_LogicalDisk, 52

CIM_LogicalDiskBasedOnVolume, 52

CIM_MediaAccessDevice, 53

CIM_MediaPresent, 52

CIM_NFS, 52

CIM_ParallelController, 52

CIM_Processor, 52

CIM_ResidesOnExtent, 52

CIM_SCSIController, 52

CIM_SCSIInterface, 53

CIM_Service, 53

CIM_SoftwareElement, 54

CIM_StorageVolume, 53

CIM_UnixComputerFileSystem, 53

CIM_UnixLocalFileSystem, 53

CIM_UnixOperatingSystem, 53

CLASS attribute, 63, 67

clean machine, definition, 13

Client Methods Class, 45

CMETHOD Class, 45

CNDITION attribute, 63, 67

COLLECT attribute, 48

command line

installing

agent for Linux, 27

agent for UNIX, 36

Common Information Model, 16

definition, 13

schema, 52

CONNECT configuration file attribute, 34

copyright notices, 2

CPU type, 73

CRC, 68

resource, 68

service, 74

CREATE method, 102

customer support, 5

D

daemons.sh, sample shell scripts, 40

DAILY ZSCHDEF variable, 105

DATE attribute, 60

DEFERRED, 100

Delete Discovered Application Component, 77

DELETE method, 102

DEPTH instance, 47

DESCRIPT attribute, 72
DESKTOP Class, 45
DIFF attribute, 62
Differenced Audit on Install and Update, 87
Differenced Audit on Install, Verify, and Update, 88
DIR instance, 47
DIRPATH attribute, 60
DISPLAY attribute, 24, 27
Distributed Management Task Force, 16
DMI, 16
DMTF. *See* Distributed Management Task Force
DMTF Web site, 54
document changes, 4
documentation updates, 4
DOMAIN attribute, 101
drive

- assigned letters, 73
- current free space, 73
- free space, 73
- mount, 73
- name, 73
- total space, 73

E

EXCLUDE instance, 47

F

FILE class, 45
File Scanner class, 45
File Scanner Filters, 46
FILEAUDT object, 58, 59

- ACCESSDT, 60
- ACCESSTM, 60
- DATE, 60
- DIRPATH, 60
- FULLPATH, 60
- PAHTCRC, 60
- SIZE, 60
- TIME, 60

FILESCAN class, 45
filescan.tkd, 61, 67
FILTER class, 46
filtering patch reports, 117
free drive space, 73
Full Audit on Install and Update, 88
Full Audit on Install, Verify and Update, 88
FULLPATH attribute, 60

G

graphical installation

- agent, 28

graphical user interface

- installing agent
 - local computer, 28
 - remote computer, 31

GROUP attribute, 56

H

HARDWARE keyword, 68
HOME directory, 25
host ID, 73
host name, 74
HOURLY ZSCHDEF variable, 106
HPCA agent

- description, 13

HPCA Core, 12
HPCA Satellite, 12

I

IDE controllers, 53
IDMLIB configuration file attribute, 33
IDMLIB directory, 37
IDMLOG configuration file attribute, 33
IDMSYS location, 25
IMMEDIATE, 100
INCLUDE instance, 47

- Individual File Audit, 77
- installation methods
 - agent, 27
- installation program
 - agent, 40
- installing
 - agent
 - using command line, 36
 - agent
 - using GUI
 - local computer, 28
- Installing the HPCA Agent to a Thin Client, 38
- INSTDIR configuration file attribute, 33
- INTERNAL attribute, 57
- INTERVAL ZSCHDEF variable, 106
- Inventory Options Class, 46, 47
- inventory reports, 114
- Inventory Scanners class, 46
- inventorydata, 19
- inventorytemp, 20
- IP address, 73
- IPADDR attribute, 101
- IPADDR01 attribute, 72

L

- LADAPT01 attribute, 73
- LAN adapter, 73
- LAN number, 73
- LANGUAGE attribute, 57
- LANNUM attribute, 73
- legal notices
 - copyright, 2
 - restricted rights, 2
 - warranty, 2
- library files, 25
- Linux environment variables, 25
- LOCATION attribute, 58

- LogicalDiskBasedOnExtent, 52
- loglevel 9, 41

M

- MediaAccessDevice, 52
- Messaging Server, 13
- MGRIP configuration file attribute, 34
- MGRNAME attribute, 102
- MGRPORT configuration file attribute, 34
- MGRVLOC class, 46
- MGRVLOC instance, 57
- Microsoft SQL server, 18
- mode plain, 27
- mode silent, 36
- mode text
 - agent installation, 27
- MOTIF, 25

N

- NAME attribute, 48, 58, 61, 63, 68, 101
- NAME instance, 47
- NAMESPACE attribute, 63, 67
- network adapter IP address, 72
- non-graphical installation
 - agent, 36
- Notify, 40
- NTFYPORT configuration file attribute, 34
- NUMDAYS ZSCHDEF variable, 106
- nvdcm.tkd, 16, 91
- NVDM Discovery of Applications, 77
- nvdrc file, 25

O

- object child ID, 68
- OBJECTS configuration file attribute, 34
- operating system revision number, 73

operating system version, 73
ORGNAME attribute, 57
OSREV attribute, 73
OSVER attribute, 73
OUTPUT attribute, 56, 62, 63, 67

P

PACKAGE Class, 45, 86, 88
Palm PDA XML Inventory, 77
parent class ID, 68
parent class name, 68
PATH class, 46
PATHCRC attribute, 60
PENDING attribute, 107
PERIODIC, 100
plain mode for installing
 agent, 27
Portal, description, 14
POST_WBEM method
 disabling, 69
POSTPARAM configuration file attribute, 34
POSTPROC configuration file attribute, 34
PREPARAM configuration file attribute, 34
PREPROC configuration file attribute, 33
PRIMARY.AUDIT, 76
PRODUCT attribute, 57
PROVERS attribute, 57
PROFILE File, 26
PROPERTY attribute, 63, 67

Q

QMSG.EXE, 69

R

radexecd, 40
RADIA_UPLOAD, 47

radsched, 40
RAM total memory, 73
Red Hat Linux Prerequisites, 25
REGISTRY class, 46
REMOTE configuration file attribute, 33
Remote Installation Setup, description, 26
Reporting Server, 14
 filtering patch reports, 117
Reporting Server, 12
 overview, 12
REQUEST attribute, 101
resolution return code, 74
resolution type, 74
resource CRC, 68
resource size, 68
restricted rights legend, 2
RIM Reporting, 78
RIM_REPORTING service, 110
RIMDIFF method, 54, 55
RIMFSCAN method, 54
RIMOPTS, 93
 customizing, 109
 Default, 87
 Differenced Audit on Install and Update, 87
 Differenced Audit on Install, Verify, and Update,
 88
 Full Audit on Install and Update, 88
 Full Audit on Install, Verify and Update, 88
RIMOPTS Class, 46, 47, 87
 attributes, 48
RIMOPTS instance, 48
RIMWBEM method, 62
RUNEXEC attribute, 48, 110

S

sample shell scripts, 40
SCANFOR attribute, 55

- SCANNER class, 46
- Scheduler, 40
- Scheduling Class, 46, 98
 - _ALWAYS_, 101
 - APPSVC, 101
 - attributes, 100
 - DOMAIN, 101
 - IPADDR, 101
 - MGRNAME, 102
 - NAME, 101
 - REQUEST, 101
 - SOCKET, 101
 - ZCREATE, 102
 - ZDELETE, 102
 - ZNOPING, 102
 - ZOBJPRI, 100
 - ZRSCCMDL, 101
 - ZSCHDEF, 100
 - ZSCHFREQ, 101
 - ZSCHMODE, 100
 - ZSCHTYPE, 100
 - ZSTOP, 100
 - ZSVCOID, 101
 - ZUPDATE, 102
 - ZVERIFY, 102
- Scheduling instance, creating, 103
 - service class, 74
 - service CRC, 74
 - service date, 74
 - service domain, 74
 - service name, 74
 - service parent ID, 74
 - service time, 74
 - shell scripts, examples, 40
 - silent installation, 34, 35
 - command line parameters, 36
 - SIZE attribute, 60
 - size of resource, 68
 - SNMP, 16
 - SOCKET attribute, 101

- SQL Inventory database, creating, 19
- SQL Inventory Manager database, creating, 19
- subscriber, definition, 14
- support, 5
 - system requirements
 - agent, 24
- SYSTEM.PROCESS.WBEMAUDT
 - disabling, 69
- SYSTEM.ZMETHOD.POST_WBEM
 - disabling, 69

T

- tablespace, creating, 19
- technical support, 5
- text mode for installing
 - agent, 27
- thin client, 38
- TIME attribute, 60
- TIMER Class, 46, 98
 - _ALWAYS_, 101
 - APPSVC, 101
 - attributes, 100
 - DOMAIN, 101
 - IPADDR, 101
 - MGRNAME, 102
 - NAME, 101
 - REQUEST, 101
 - SOCKET, 101
 - ZCREATE, 102
 - ZDELETE, 102
 - ZNOPING, 102
 - ZOBJPRI, 100
 - ZRSCCMDL, 101
 - ZSCHDEF, 100
 - ZSCHFREQ, 101
 - ZSCHMODE, 100
 - ZSCHTYPE, 100
 - ZSTOP, 100
 - ZSVCOID, 101
 - ZUPDATE, 102
 - ZVERIFY, 102

TIMER instance, creating, 103
timer settings, 105
timer-based deployments, 40
total drive space, 73
TYPE attribute, 56, 63, 67

U

Unix File Audit methods, 93
 creating a package, 93
Unix File Scan Audit, 78
Unix Hardware Inventory, 78
Unix Software Inventory Audit, 78
UPDATE method, 102
updates to doc, 4
user ID, 74

V

VENDOR attribute, 57
VERIFY method, 102
VERSION attribute, 57
Virtual Mgr Location class, 46

W

warranty, 2
WBEM. *See also* Web-Based Enterprise Management
 definition, 14
 URL for more information, 15
WBEM auditing, 62
WBEM Class, 46, 67
WBEM MSI Based Applications, 78
WBEM namespace, 16, 62, 65, 67
WBEM object
 attributes, 64, 68
WBEM object variables
 ZOBJCID, 64
 ZOBJCLAS, 64
 ZOBJCRC, 64

ZOBJDATE, 64
ZOBJDOMN, 64
ZOBJID, 64
ZOBJNAME, 64
ZOBJPCLS, 64
ZOBJPID, 64
ZOBJRCRC, 64
ZOBJRSIZ, 64
ZOBJTIME, 64
ZRSCSRC, 64

WBEM Object Variables

ZOBJCID, 68
ZOBJCLAS, 68
ZOBJCRC, 68
ZOBJDATE, 68
ZOBJID, 68
ZOBJNAME, 68
ZOBJPCLS, 68
ZOBJPID, 68
ZOBJRCRC, 68
ZOBJRSIZ, 68
ZOBJTIME, 69
ZRSCSRC, 69

WBEM Running Services, 79

WBEM Scan for Hardware, 79

WBEM Scan with Condition Statement, 79

WBEM Stopped Services, 79

WBEM System Drivers, 79

WBEM Windows Services, 79

WBEMAUDT object, 82
 processing, 69

WBEMAUDT process
 disabling, 69

WBEMCURR object, 82

WBEMPREV object, 82

Web-Based Enterprise Management, 15
 definition, 14

WEEKDAY ZSCHDEF variable, 106

WEEKLY ZSCHDEF variable, 105

Windows Registry, scanning, 46

Windows System DLL, 79
WMI used for registry scan, 46

X

xmlCIM, 16

Z

ZCONFIG object, 70
 attributes, 72

ZCONFIG variable, 71

ZCRCINFO attribute, 58

ZCREATE attribute, 102

ZDELETE attribute, 102

ZHDWCPU attribute, 73

ZHDWD00 attribute, 73

ZHDWD00F attribute, 73

ZHDWD00M attribute, 73

ZHDWD00T attribute, 73

ZHDWD01 attribute, 73

ZHDWD01F attribute, 73

ZHDWD01M attribute, 73

ZHDWD01T attribute, 73

ZHDWD02 attribute, 73

ZHDWD02F attribute, 73

ZHDWD02M attribute, 73

ZHDWD02T attribute, 73

ZHDWD03 attribute, 73

ZHDWD03F attribute, 73

ZHDWD03M attribute, 73

ZHDWD03T attribute, 73

ZHDWDNUM attribute, 73

ZHDWIPAD attribute, 73

ZHDWLANA attribute, 73

ZHDWMEM attribute, 73

ZHDWOS attribute, 73

ZHDWXHID attribute, 73

ZHDWXHN attribute, 74

ZINIT attribute, 58

ZNOPING attribute, 102, 107

ZOBJCID attribute, 64, 68

ZOBJCLAS attribute, 64, 68

ZOBJCRC attribute, 64, 68

ZOBJDATE attribute, 64, 68

ZOBJDOMN attribute, 64, 68

ZOBJID attribute, 64, 68

ZOBJNAME attribute, 64, 68

ZOBJPCLS attribute, 64, 68

ZOBJPID attribute, 64, 68

ZOBJPRI attribute, 100

ZOBJRCRC attribute, 64, 68

ZOBJRRC attribute, 74

ZOBJRSIZ attribute, 64, 68

ZOBJRSTY attribute, 74

ZOBJTIME attribute, 64, 69

ZRSCCMDL attribute, 101, 106, 107

ZRSCCRC attribute, 58

ZRSCMFIL attribute, 57

ZRSCMMEM attribute, 57

ZRSCOBJN attribute, 58

ZRSCPADM attribute, 58

ZRSCSRC attribute, 58, 64, 69

ZRSCSTYP attribute, 57

ZRSCVLOC attribute, 55, 57

ZRSCVLOC instance, 47

ZSCHDEF attribute, 100, 105

ZSCHFREQ attribute, 101, 106

ZSCHMODE attribute, 100

ZSCHTYPE attribute, 100, 106

ZSERVICE Class, 45

ZSERVICE instance
 creating, 92
ZSRCCLAS attribute, 74
ZSRCCRC attribute, 74
ZSRCDATE attribute, 74
ZSRCDOMN attribute, 74
ZSRCNAME attribute, 74
ZSRCPID attribute, 74
ZSRCTIME attribute, 74
ZSTOP attribute, 100
ZSTOP, updating, 92
ZSVCMO attribute, 110
ZSVCOID attribute, 101
ZSVCTYPE attribute, 48, 87
ZTASKEND rexx, 69
ZTIMEQ object, 106, 107
ZUPDATE attribute, 102
ZUSERID attribute, 74
ZVERIFY attribute, 102
ZVERINFO attribute, 56

