HP OpenView Configuration Management

Inventory Manager

for UNIX operating systems

Software Version: 5.00

Installation and Configuration Guide

Document Release Date: April 2007

Software Release Date: April 2007

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Trademark Notices

DHTML Calendar

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version
- 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:

ovweb.external.hp.com/lpe/doc_serv/

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 lists new features added for the Configuration Management v 5.0 release.

Table 1 New features added for Configuration Management 5.0

Chapter	Version	Changes
2	5.00	Page 19, Creating the CM Inventory Manager Environment: new chapter explains how to define an ODBC database and DSN for CM Inventory data, and how to setup the various CM Infrastructure servers used to collect (CM Configuration Server), post (CM Messaging Server), and report (CM Reporting Server) the data.
3	5.00	Page 26, System Requirements: Updated System Requirements for version 5.0.
3	5.00	Page 30, To install the CM agent: Updated installation: • New panel to prompt user when to start CM Daemons. • New panel to prompt for WBEM server libraries link creation and search path.
4	5.00	Page 66, Manual Scanning Using RIMWBEM: Added instructions for running RIMWBEM from the command line.
4	5.00	Page 71, WBEM Object : section updated to include new methods for posting audit data using CM Messaging Server.

Chapter	Version	Changes
10	5.00	Page 119, Viewing Inventory from the CM Reporting Server: new chapter for viewing Inventory reports. Replaces the earlier chapter: “Viewing Inventory from the Radia Integration Server”.

Table 2 Document changes

Chapter	Version	Changes
4	4.0	Registry class information added to AUDIT Domain table.
1	4.0	Added CM Messaging Server and CM Reporting Server information to Introduction.

Support

You can visit the HP Software support web site at:

www.hp.com/managementsoftware/services

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:

www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

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

Contents

1 Introduction

At the end of this chapter, you will:

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

About this Guide

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

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

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

About CM Technology

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

- **The Embedded Intelligence of Object-Oriented Technology**
  Object-oriented technology transforms software and content from file-based media into self-aware, platform-independent, intelligent objects that automatically assess the environment into which they are deployed, and personalize, install, update, and repair themselves accordingly. In other words, as intelligent objects, they know what they need for a particular device or user, where to get what they need, when they need to change, how to change themselves, and how to repair themselves.
- **Revisable Packaging for Revisable Content**
  CM technology enables revision and customization of software and content at any midstream point in the publisher-to-subscriber deployment process. Because CM technology transforms software and content into objects, these objects can be easily modified midstream – subtracted from, added to, reconfigured – simply by packaging them with

other objects or new configuration information. With revisable packaging, value-added service providers and IT administrators can customize standard published software offerings for the needs of their particular users without having to unpack and repackage everything.

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

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

About the CM Reporting Server

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

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

Refer to the *CM Reporting Server Guide* for more information on how to install the CM Reporting Server and create a reporting environment for your SQL databases for Configuration Management.

Terminology

agent computer

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

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.

CM agent

The **Configuration Management agent** is the CM software component that is installed on the end user's desktop computer. There are CM agents for the CM Application Manager, the CM Application Self-service Manager, the CM Inventory Manager, the CM Patch Manager, and the CM OS Manager.

CM Portal

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

CM Messaging Server

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

CM Reporting Server

The CM Reporting Server is a Web-based interface to the reportable data captured by the CM extended infrastructure product suite. It allows you to query the combined data in existing CM Inventory Manager, CM Patch Manager, and CM 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.

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.

subscriber

A **subscriber** is the person (end user) who uses CM -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.

CM Prerequisites

The CM Inventory Manager 5.00 requires the following CM components:

- CM Configuration Server 5.00 or higher
- CM agent 5.00 or higher
  - CM Application Manager

  and/or

  - CM Application Self-service Manager
- CM Messaging Server 5.00. Refer to the *CM Messaging Server Guide* for more information on installing or migrating to the CM Messaging Server, and how the CM Messaging Server transfers data directly, or indirectly, to CM databases.

Hardware/Software

Use the CM Admin CSDB Editor to manipulate the CM Configuration Server DB. The CM Admin CSDB Editor is part of the CM Administrator and is available for Windows platforms. Install the CM Administrator onto a Windows computer that has access to your CM Configuration Server DB.

Necessary Skills

With CM Products

This document assumes that the reader is familiar with the CM Configuration Server DB and with administering CM using the CM Admin CSDB Editor. Refer to the *CM Admin CSDB Editor 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`

CM Inventory Manager Technology

While an administrator with little web-based knowledge can use the CM 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 CM 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).

CM and WBEM

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

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

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

Summary

- The CM Inventory Manager is a utility used to discover configuration on remote computers.
- The collection of inventory information occurs on the CM Inventory Manager when a subscriber connects to the CM Configuration Server.
- The CM Messaging Server and CM Reporting Server are used to collect and report on CM data stored in a SQL Server or Oracle database.
- We suggest that the administrator be familiar with HP OpenView products as well as Web-Based Enterprise Management (WEBM), and Microsoft's implementations of WBEM.
- All information collected by WBEM is available to the CM Inventory Manager.
- The CM Inventory Manager queries the WBEM namespace (i.e., the WBEM database) and sends the results back to the CM Configuration Server.

2 Creating the CM Inventory Manager Environment

At the end of this chapter, you will:

- Be familiar with the tasks needed to set up the CM infrastructure and SQL or Oracle database for a CM Inventory Manager environment.
- Create a SQL or Oracle database and ODBC DSN connection for the CM Inventory Manager.
- Have the required infrastructure installed and configured to support a CM Inventory Manager environment.

CM Inventory Manager Implementation Tasks

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

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

SQL or Oracle Prerequisite

Before setting up your environment for the CM 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 CM Inventory Manager

Before installing CM 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.

➤ 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: CMAUDIT (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 CM Configuration Server and co-located CM Messaging Server, create an ODBC DSN called CMINVMGR, 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 inventory`data.dbf`
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 `inventorytemp.dbf`
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	cminventory
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 CM Configuration Server and CM Messaging Server, create an ODBC DSN called CMINVMGR, 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 CM 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.

CM Administrator CSDB Editor

The CM media contains a CM Administrator installation, which allows you to install several tools. Use it to install the CM Admin CSDB Editor. Refer to the *CM Application Manager Guide* or the *CM Application Self-service Manager Guide* for more information on installation. Instructions for using the CM Admin CSDB Editor can be found in the *CM Administrator CSDB Editor Guide*.

➤ The CM Admin CSDB Editor is available for Windows platforms only.

CM Messaging Server

Install CM Messaging Server 5.00 on the CM 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 CM Inventory Manager-related data to the appropriate back-end CM Inventory Manager database:

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

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

CM Reporting Server

The CM Reporting Server 5.00 is required to view enhanced reports for CM Inventory Manager. Review the CM Reporting Server release notes prior to installing. The *CM Reporting Server Guide* also includes instructions on how to use its flexible features.

CM Portal (Optional)

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

Summary

- Install the CM Configuration Server and its database. Also install the CM Admin CSDB Editor.
- CM Inventory Manager requires an SQL or Oracle database and an ODBC DSN connection to the data source.
- Install the CM Messaging Server on the CM Configuration Server. Include the three DDAs related to CM Inventory Manager data: CORE.DDA, INVENTORY.DDA and WBEM.DDA. Configure each of these DDAs to post its data using ODBC to the desired ODBC database.
- Install the CM Reporting Server and configure it to access your CM Inventory Manager database.
- Optionally, install the CM Portal, which offers remote installations of any CM agent, including the CM Inventory Manager Agent.

3 Installing the CM Agents

At the end of this chapter, you will:

- Understand the system requirements and permissions necessary to deploy the CM agent.
- Be able to install the CM agent using either the graphical or non-graphical mode.

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

System Requirements

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

Platform Support

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

Prerequisites

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

Table 3 [PROPERTIES] Section of INSTALL.INI

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

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

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

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

Table 4 Environment Variables

Platforms	Examples

Solaris	`LD_LIBRARY_PATH=/lib:$IDMSYS:$MOTIF:$LD_LIBRARY_PATH`
	`PATH=/bin:/usr/bin:$IDMSYS:$MOTIF:$PATH`

HP-UX	`SHLIB_PATH=/lib:$IDMSYS:$MOTIF:$SHLIB_PATH`
	`PATH= /bin:/usr/bin:$IDMSYS:$MOTIF:$PATH`

AIX	`LIBPATH=/lib:$IDMSYS:$MOTIF:$LIBPATH`
	`PATH=/bin:/usr/bin:$IDMSYS:$MOTIF:$PATH`

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

In Table 4 above, `$IDMSYS` represents the fully-qualified path to the CM 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 CM agent or CM Administrator graphical tools such as the CM Admin Publisher, the CM Admin Agent Explorer, and the presentation of the CM agent logon panel.

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

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

Recommendations

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

Before you install the CM agent, consider the following:

- You can perform a local installation of the CM agents.
- Your CM 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 CM agent installation.
  - Ability to use a post-installation method, which runs any script or executable after the CM agent is installed.
  - You can configure the installation to force an agent connection to the CM Configuration Server immediately after the installation.
  - You can pre-configure the IP address and port number of the CM Configuration Server that the CM agent will be connecting to.
  - Ability to use an object update text file that can be used to update CM objects after the installation.

Installation Methods

You can install the CM agents by:

- Executing the installation procedure directly from the CM media.

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

Several parameters can be used on the command line when installing the CM agents. These parameters are used to install the CM agent using the graphical mode, non-graphical mode, plain mode, or silent mode. Table 5 below, describes the installation parameters.

Table 5 Command Line Installation Parameters

Parameter	Example	Description
`-mode plain`	`./install -mode plain`	Installs the CM agent in plain mode. The installation graphics are displayed with no animations. This is useful for remote installations where network bandwidth may be an issue.
`-mode text`	`./install -mode text`	Installs the CM 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 more 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 CM agent

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

In order for CM to install correctly on HP-UX platforms, you must mount the media using pfs_mount.

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

Insert the CM media and mount by typing:

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

where */dvdrom* is your physical media device.

To un-mount, type:

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

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

Graphical Installation

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

To install the CM agent to a local computer using a GUI

These instructions will guide you through the local graphical installation of the CM agent. For the non-graphical installation instructions, see Non-graphical Installation on page 41.

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

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

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

3 The Welcome window opens.

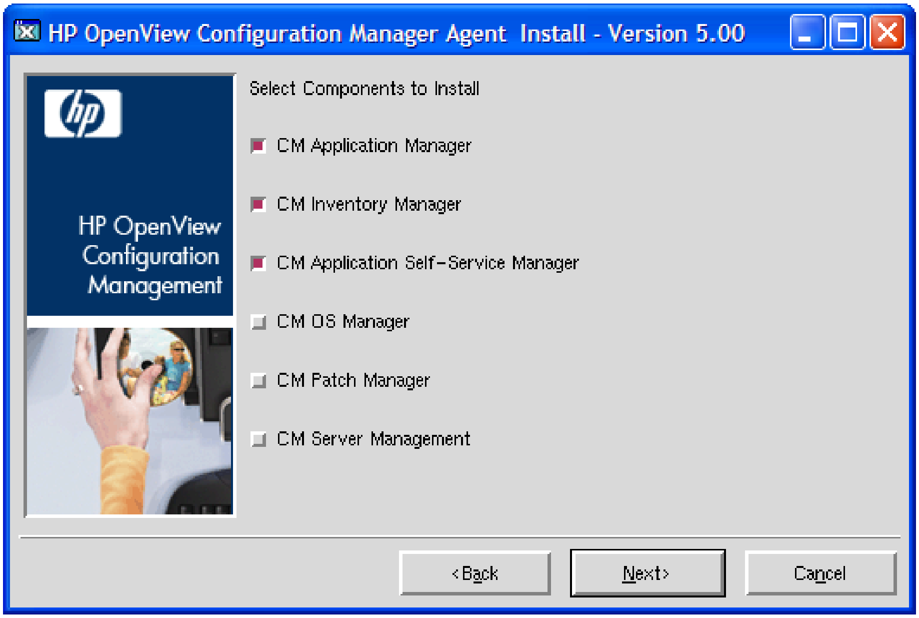
> At any point during the installation, you can return to a previous window by clicking **Back**. Also, if you would like to exit the installation at any time, click **Cancel**.

4 Click **Next**.

The End User License Agreement window opens.

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

The Select Components to Install window opens.



HP OpenView Configuration Manager Agent Install - Version 5.00

Select Components to Install

☑ CM Application Manager

☑ CM Inventory Manager

☑ CM Application Self-Service Manager

□ CM OS Manager

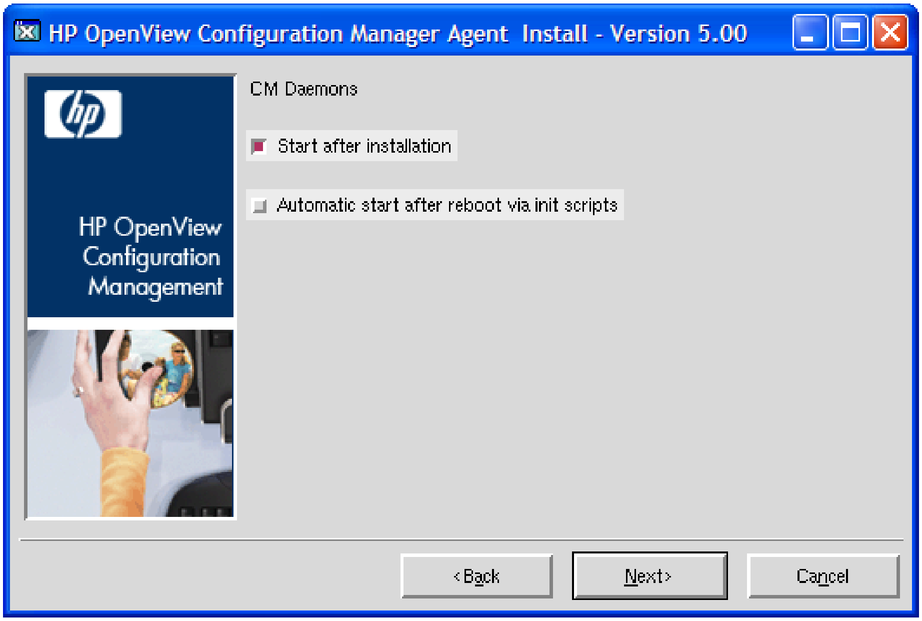
□ CM Patch Manager

□ CM Server Management

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

7 Click **Next**.

The CM Daemons window opens.


HP OpenView Configuration Manager Agent Install - Version 5.00
HP OpenView Configuration Management
< Back
Next >
Cancel

CM Daemons

☒ Start after installation

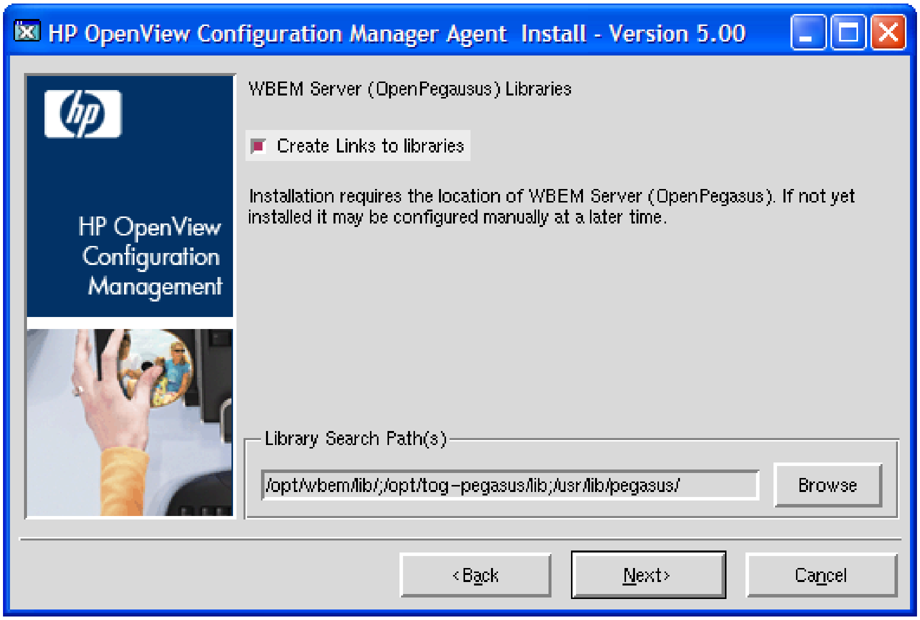
□ Automatic start after reboot via init scripts

8 Select when you want the CM Daemons to start. The CM Daemons run on the agent computer and perform CM management tasks. See About CM Daemons in UNIX on page 42 for more 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.

9 Click **Next**.

The WBEM Server (OpenPegasus) Libraries window opens. If you are running the installation on a Solaris device, you will be prompted for CIM server login credentials, see below.



HP OpenView Configuration Manager Agent Install - Version 5.00
HP OpenView Configuration Management
< Back
Next >
Cancel
Browse

WBEM Server (OpenPegausus) Libraries

☑ Create Links to libraries

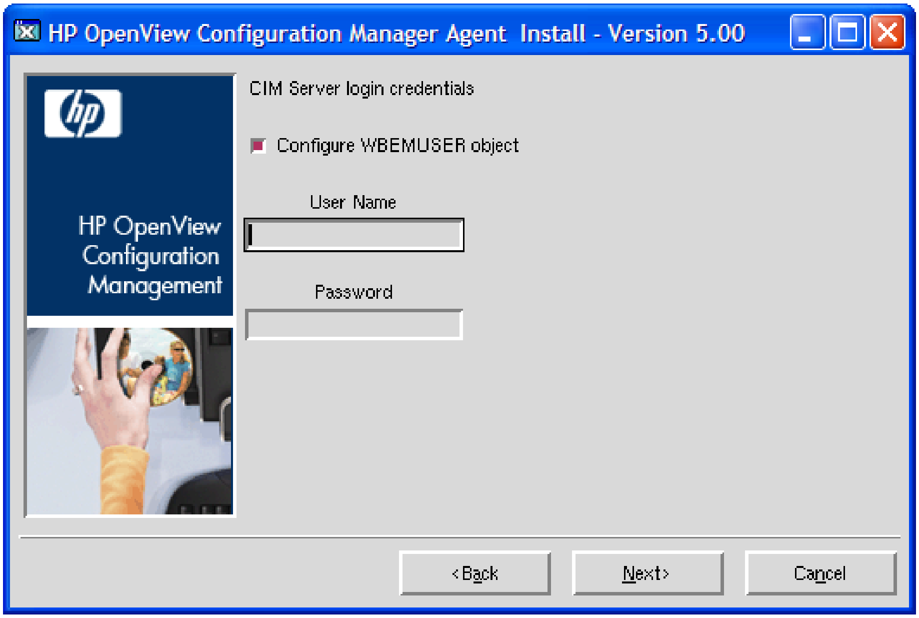
Installation requires the location of WBEM Server (OpenPegasus). If not yet installed it may be configured manually at a later time.

Library Search Path(s)

/opt/wbem/lib/;/opt/tog-pegasus/lib;/usr/lib/pegasus/

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 CM agent is installed.

If you are running the installation on a Solaris device, the CIM Server login credentials window opens.



HP OpenView Configuration Manager Agent Install - Version 5.00
HP OpenView Configuration Management
< Back
Next >
Cancel

CIM Server login credentials

☑ Configure WBEMUSER object

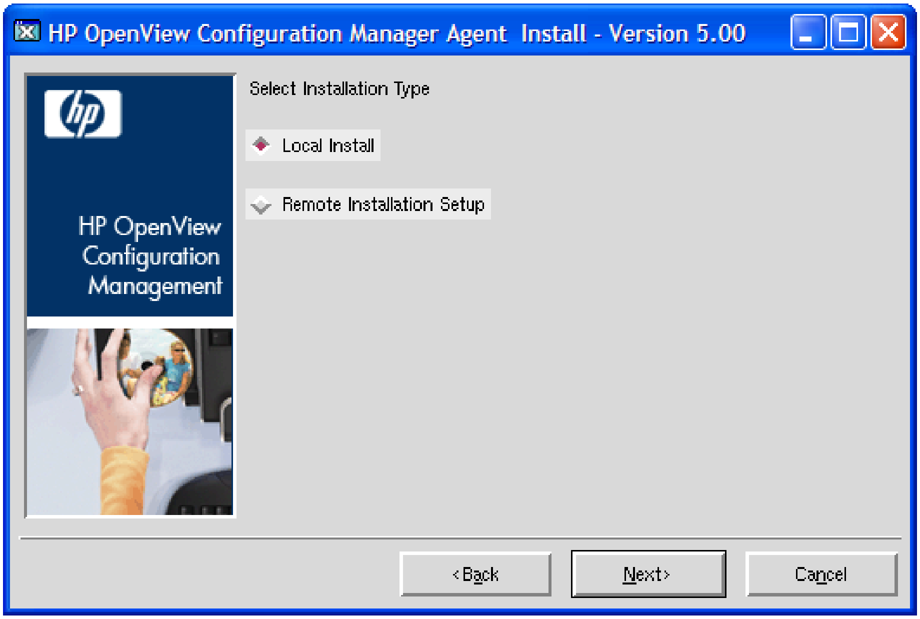
User Name

Password

10 Select to configure the WBEMUSER object by adding a user name and password (Solaris only).

11 Click **Next**.

The Select Installation Type window opens.



HP OpenView Configuration Manager Agent Install - Version 5.00
HP OpenView Configuration Management

Select Installation Type

● Local Install

○ Remote Installation Setup

12 Select **Local Install** to install the CM agent onto a local computer, and then click **Next**.

The CM Agent Location window opens.

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

14 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.

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

16 Click **Next**.

The Log Directory window opens.

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

18 Click **Next**.

The CM Configuration Server IP Address window opens.

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

20 Click **Next**.

The CM Configuration Server Port Number window opens.

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

22 Click **Next**.

The Package Settings window opens.

23 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.

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

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

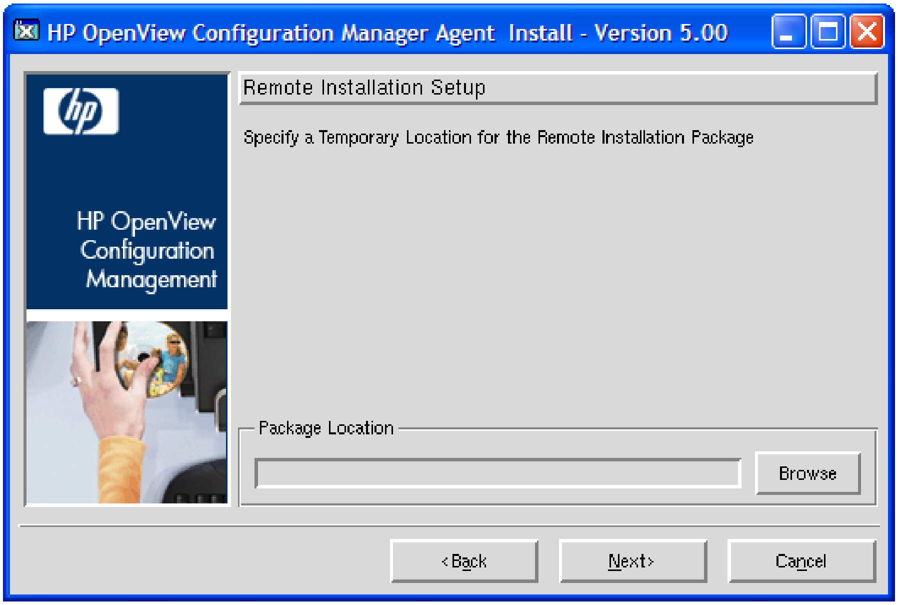
The CM agent has been successfully installed.

Remote Installation Setup

This section describes how to create a CM Agent installation configuration file that can be used to install the CM 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 more 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.



HP OpenView Configuration Manager Agent Install - Version 5.00

Remote Installation Setup

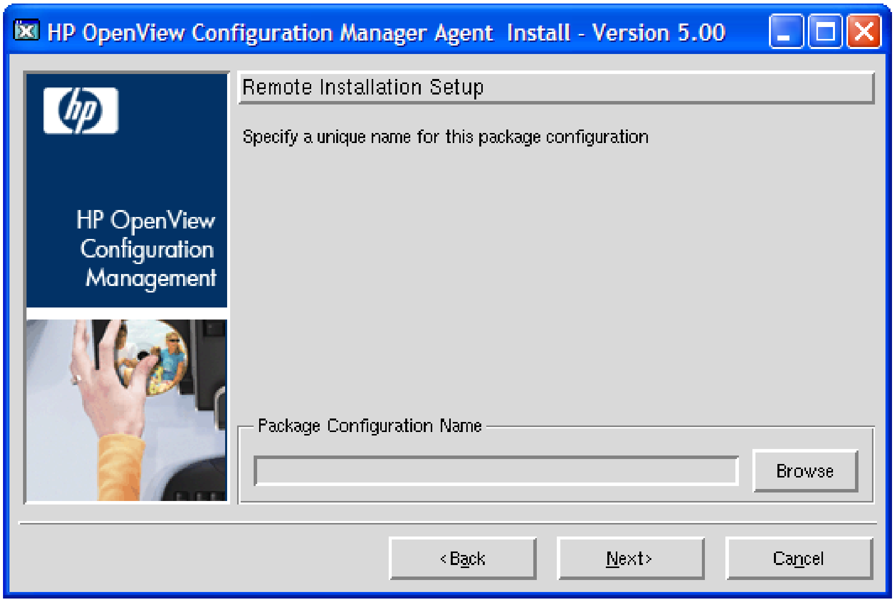
Specify a Temporary Location for the Remote Installation Package

Package Location

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

— Click **Next**.

The Package Configuration Name window opens.



HP OpenView Configuration Manager Agent Install - Version 5.00

Remote Installation Setup

Specify a unique name for this package configuration

Package Configuration Name

- 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 CM 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 CM agent installations. These responses would normally be provided during an interactive CM agent installation. When performing silent installations, more installation options are also available in the configuration file.

The variables available in the configuration file are described in Table 6, below.

Table 6 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	The default installation directory.
IDMLOG	/opt/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	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.
POSTPARM		Any parameters required by the post-installation method specified in the variable POSTPROC.
MGRIP	192.168.123.40	The default IP address for connection to the CM Configuration Server.
MGRPORT	3464	The default port number for connection to the CM Configuration Server.
NTFYPORT	3465	The default Notify port used.
CONNECT	Y	Connects to the CM Configuration Server immediately after the installation. Default behavior is N. Set to Y if you want your CM agent to connect to the CM Configuration Server automatically after the installation.
OBJECTS	./object.txt	The file that is used to create or update CM attributes after the installation.
DUAL	1	0 designates RAM only selected. 1 designates more than one component selected.

Using a Pre- or Post-Installation Script

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

```
#!/bin/sh
#
cd /opt/HP/CM/Agent

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

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

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

Customizing Installed Object Variable Content

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

```
OBJECT_NAME VARIABLE_NAME VARIABLE_VALUE
```

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

```
ZMASTER ZTRACE N
```

```
ZMASTER ZTRACEL 000
```

When creating an object text file:

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

We recommend the agent be installed as root.

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

- your CM 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 CM agent. Table 7 below describes these.

Table 7 Silent installation command line parameters

Parameter	Example	Description
`-cfg`	`./install -cfg` *`install.cfg`*	The file name specified after -cfg is the name of the configuration file to be used during the installation. For information about configuration files, see Customizing the Installation Configuration File on page 37.
`-mode silent`	`./install -mode silent -cfg` *`install.cfg`*	Installs the CM agent in silent mode based on the parameters set in the configuration file specified after the `-cfg` parameter. For information about configuration files, see Customizing the Installation Configuration File on page 37.

Non-graphical Installation

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

To install the CM agent for UNIX using a command line

These instructions guide you through the local non-graphical installation of the CM agent for UNIX. For the graphical installation, see Graphical Installation on page 30.

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

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

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

  The CM 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.

  - CM Application Manager
  - CM Inventory Manager
  - CM Application Self-service Manager
  - CM OS Manager
  - CM Patch Manager
  - CM Server Management

6 CM Daemons, press **Enter** to start the CM Daemons after install or type N and press **Enter** to start them later.

7 Automatic start after reboot via init scripts, press **Enter** to not start the CM Daemons each time the device is restarted or type Y and press **Enter** to allow CM Daemons to automatically start when the device is rebooted.

8 If you are installing to a Solaris device, you will be prompted to configure the WBEMUSER object. If you select Y you will then be prompted to supply a user name and password for the WBEMUSER object.

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

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

or

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

For this example, we accepted the default.

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

11 Specify the location for the CM proprietary objects (`IDMLIB`), and then press **Enter**.

12 Specify the location for the log files created by CM (`IDMLOG`), and then press **Enter**.

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

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

15 Review the installation settings you have chosen.

16 If you would like to install the CM 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.

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

The CM agent is installed.

About CM Daemons in UNIX

The CM agent installation program installs the following daemon executables:

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

- **CM Scheduler**
  Use the CM Scheduler service, **radsched**, to schedule timer-based application deployments.

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

Sample Shell Scripts

The installation of the CM 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 CM 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 UNIX user ID who ran the install.

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

Summary

- We strongly recommend that you install and run the CM agents as root.
- The CM agents can be installed using either the graphical or non-graphical modes.

4 The AUDIT Domain

At the end of this chapter, you will:

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

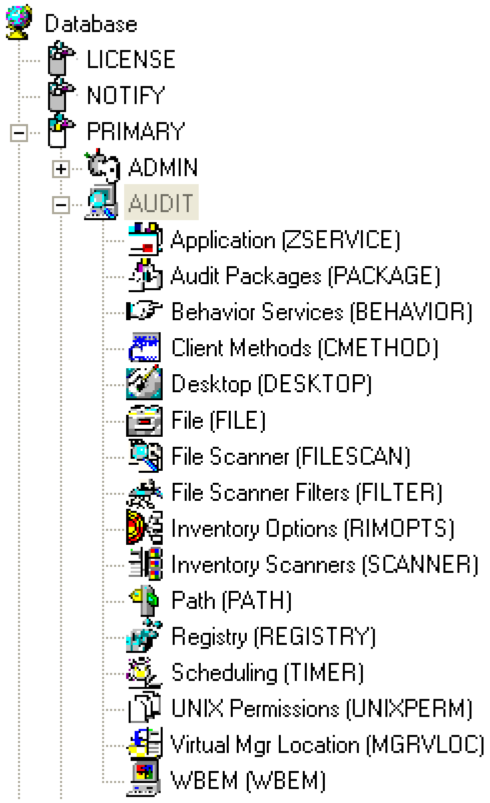
The AUDIT Domain

The AUDIT Domain is located in the PRIMARY File of the CM-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 CM Admin CSDB Editor, which is available for 32-bit Windows platforms. For more information, refer to the *CM Admin CSDB Editor Guide*.

Figure 1 PRIMARY.AUDIT Domain



```
Database
  LICENSE
  NOTIFY
⊟ PRIMARY
  ⊞ ADMIN
  ⊟ AUDIT
      Application (ZSERVICE)
      Audit Packages (PACKAGE)
      Behavior Services (BEHAVIOR)
      Client Methods (CMETHOD)
      Desktop (DESKTOP)
      File (FILE)
      File Scanner (FILESCAN)
      File Scanner Filters (FILTER)
      Inventory Options (RIMOPTS)
      Inventory Scanners (SCANNER)
      Path (PATH)
      Registry (REGISTRY)
      Scheduling (TIMER)
      UNIX Permissions (UNIXPERM)
      Virtual Mgr Location (MGRVLOC)
      WBEM (WBEM)
```

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 8 Audit Domain

Class	Description
Audit Application (ZSERVICE)	Sample services distributed with the CM 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 *Registry Class* topics for more information.
Scheduling (TIMER)	Contains the instances that enable the CM 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 CM 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 9 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 CM 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 10 on page 50 describes these attributes.

Table 10 RIMOPTS Class

Attribute	Usage
COLLECT	Audit Collection Type by selecting **Diff** or **Full** - 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. 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. - 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	Indicates what actions the CM Inventory Manager will take upon connection: - 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. The default settings are **I** and **U**.
ZSVCTYPE	Contains code that is used internally by the CM Inventory Manager agent. In all cases, this value should remain **I**.
NAME	Contains the friendly name of the instance. It is the name displayed for the instance in the tree view of the CM Admin CSDB Editor.

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 CM 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 *CM Admin CSDB Editor Guide*.

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

1. Open the CM 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.

Summary

- The AUDIT Domain contains the classes required to configure the tasks needed to collect the inventory information and to manage the agent computer's assets.
- The RIMOPTS Class is also known as the Inventory Options Class. This class contains the attributes that control an inventory management task.

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 CM and is available for reporting purposes.

For a description of the CIM schema classes used, see Table 11 below.

Table 11 CIM classes

CIM Class	Description
CIM_SCSIController	Subclass of the CIM_Controller used to represent SCSI controllers.
CIM_ResidesOnExtent	Subclass of CIM_Dependancy. 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. On HP-UX, this class also collects SD products and creates the appropriate classes.
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 CM Inventory Manager for UNIX 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 12 AUDIT.FILE Class Instances

Attribute	**Description** **Examples**
SCANFOR	Indicate a fully qualified path and file name to search for. Wildcards are permitted.
ACTION	The RIMDIFF method performs actions on the files discovered on the user's computer during the Agent connect. • **Y** configures RIMDIFF to perform the action. • **N** configures RIMDIFF to not perform the action. The first four flags determine when to report that the files were found: Report on: **Initial, New, Changed, Deleted** • **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. 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** • **Send** means to send the files to the CM Configuration Server and store them in the location indicated by the ZRSCVLOC attribute (see ZRSCVLOC in this table). • **Delete** means to delete the files from the user's computer. • **Custom** means to execute the method indicated in

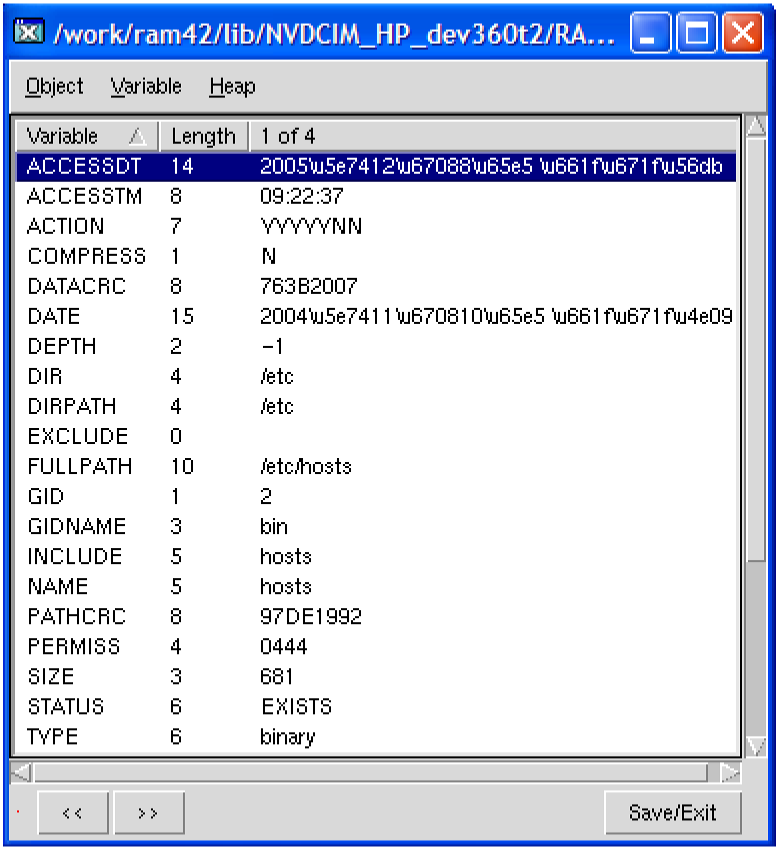
Attribute	Description Examples
	the CUSTOM attribute. YYYYNYN – Report whenever encountered and delete the files. NNYYNNN – Report when changed or deleted and take no action. NYYNYYN – Report when the files are new or changed. Then send and delete the files.
OUTPUT	Output object name.
TYPE	Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM. File.
GROUP	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.
ZVERINFO	Collect extended information. • Set the value to **1** to collect more information for a file. • Set the value to **0** to not collect more information. 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. Extended file information consists of one ore more of the following data elements. The associated attribute name for the data element is in parentheses: (VENDOR) The seller of the file/product (PRODUCT) The name of the item for which the file is a part.

Attribute	**Description** **Examples**
	(PRODVERS) The version of the product which the file is a part. (ORGNAME) The name of the organization. (INTERNAL) The internal data element encoded in the file. (VERSION) The version of the file. (LANGUAGE) The language of the file.
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 CM 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 CM 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	The product name. See ZVERINFO on page 58 for more detail.
PRODVERS	The product version. See ZVERINFO on page 58 for more detail.
ORGNAME	The organization name. See ZVERINFO on page 58 for more detail.
INTERNAL	The internal data element encoded in the file. See ZVERINFO on page 58 for more detail.
VERSION	The version of the file. See ZVERINFO on page 58 for more detail.

Attribute	Description Examples
LANGUAGE	The language of the file. See ZVERINFO on page 58 for more detail.
VENDOR	The product vendor. See ZVERINFO on page 58 for more detail.
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 CM Admin Agent Explorer to view the FILEPREV object results as shown below.

Figure 2 FILEPREV object created with RIMFSCAN



Variable	Length	1 of 4
ACCESSDT	14	2005\u5e7412\u67088\u65e5 \u661f\u671f\u56db
ACCESSTM	8	09:22:37
ACTION	7	YYYYYNN
COMPRESS	1	N
DATACRC	8	763B2007
DATE	15	2004\u5e7411\u670810\u65e5 \u661f\u671f\u4e09
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 13 **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.
DATACRC	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 CM 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 85 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 14 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 15 AUDIT.WBEM Instance

Attribute Name	Description
ACTION	RIMDIFF method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect. • **Y** configures RIMDIFF to perform the reporting action. • **N** configures RIMDIFF to not perform the reporting action. The first four flags determine *when* to report that the WBEM namespace instance was found: Report on: **Initial, New, Changed, Deleted, Scan, Delete, Custom** • **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.

Attribute Name	Description
	• **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. • **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. The last three flags are not applicable to WBEM audits.
NAMESPACE	Name of the WBEM namespace to query or HARDWARE.
CLASS	Name of the WBEM Class to query or HARDWARE.
PROPERTY	Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.
CNDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the CM 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 CM 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 CM 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 CM Configuration Server.

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

Table 16 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 CM Configuration Server.
ZOBJRSIZ	Resource size maintained by the CM 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 or Solaris cimworkshop).

WBEMUSER (Solaris Only)

The Solaris version of the Inventory Manager requires another WBEM object called WBEMUSER. This object contains two attributes, USERNAME and PASSWORD, which must both contain a valid value in order to retrieve Solaris inventory information. WBEMUSER is located by default in the `IDMROOT` directory.

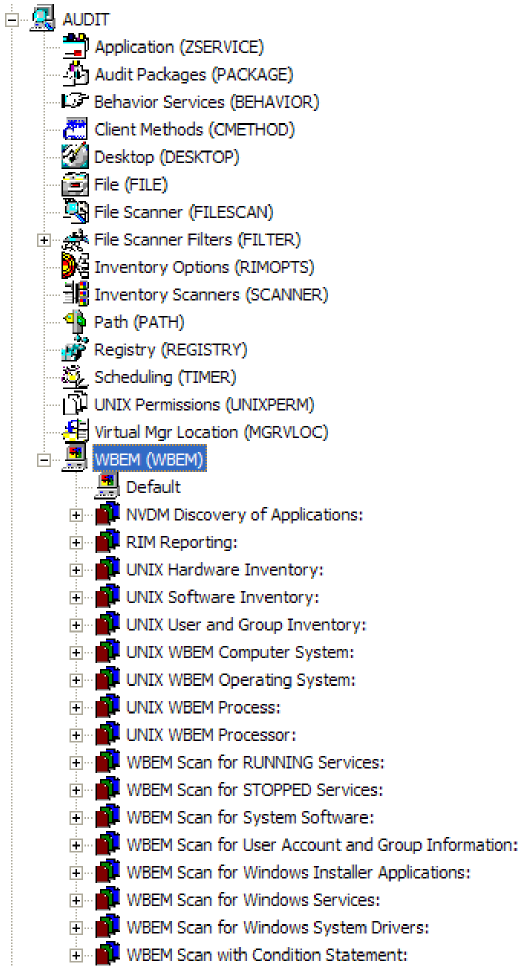
NVDCIM.TKD

UNIX WBEM auditing using `nvdcim.tkd` is supported for legacy purposes only. New installations of the CM 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



- AUDIT
  - Application (ZSERVICE)
  - Audit Packages (PACKAGE)
  - Behavior Services (BEHAVIOR)
  - Client Methods (CMETHOD)
  - Desktop (DESKTOP)
  - File (FILE)
  - File Scanner (FILESCAN)
  - File Scanner Filters (FILTER)
  - Inventory Options (RIMOPTS)
  - Inventory Scanners (SCANNER)
  - Path (PATH)
  - Registry (REGISTRY)
  - Scheduling (TIMER)
  - UNIX Permissions (UNIXPERM)
  - Virtual Mgr Location (MGRVLOC)
  - WBEM (WBEM)
    - Default
    - NVDM Discovery of Applications:
    - RIM Reporting:
    - UNIX Hardware Inventory:
    - UNIX Software Inventory:
    - UNIX User and Group Inventory:
    - UNIX WBEM Computer System:
    - UNIX WBEM Operating System:
    - UNIX WBEM Process:
    - UNIX WBEM Processor:
    - WBEM Scan for RUNNING Services:
    - WBEM Scan for STOPPED Services:
    - WBEM Scan for System Software:
    - WBEM Scan for User Account and Group Information:
    - WBEM Scan for Windows Installer Applications:
    - WBEM Scan for Windows Services:
    - WBEM Scan for Windows System Drivers:
    - WBEM Scan with Condition Statement:

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

Table 17 **AUDIT.WBEM Instance**

Attribute	Description
ACTION	The `filescan.tkd` method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the Agent connect. • `Y` configures `filescan.tkd` to perform the reporting action. • `N` configures `filescan.tkd` to not perform the reporting action. The first four flags determine *when* to report that the WBEM namespace instance was found: Report on: Initial, New, Changed, Deleted • `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. The last three flags are not applicable to WBEM audits.
NAMESPACE	Name of the WBEM namespace to query or HARDWARE.
CLASS	Name of the WBEM Class to query or HARDWARE.
PROPERTY	Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.
CNDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the CM 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 CM 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 CM 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 CM Configuration Server.

The WEBM object contains more attributes described in the table below.

Table 18 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 CM Configuration Server.
ZOBJRSIZ	The resource size maintained by the CM 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 CM Inventory Manager agent sends a WBEMAUDT object to the CM Configuration Server, processing is defined as follows:

1. When a WBEMAUDT object is found, the CM Configuration Server ZTASKEND calls QMSG.EXE.
2. QMSG.EXE places the WBEMAUDT objects into the CM Configuration Server's `\data\wbem` directory, or message queue.
3. The CM 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 CM Messaging Server located closer to the database. In the later case, the receiving CM 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 CM Reporting Server.

For more information, refer to the *CM Messaging Server Installation and Configuration Guide*.

Disabling Remnant CM Configuration Server instances for WBEM Object Processing

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

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

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

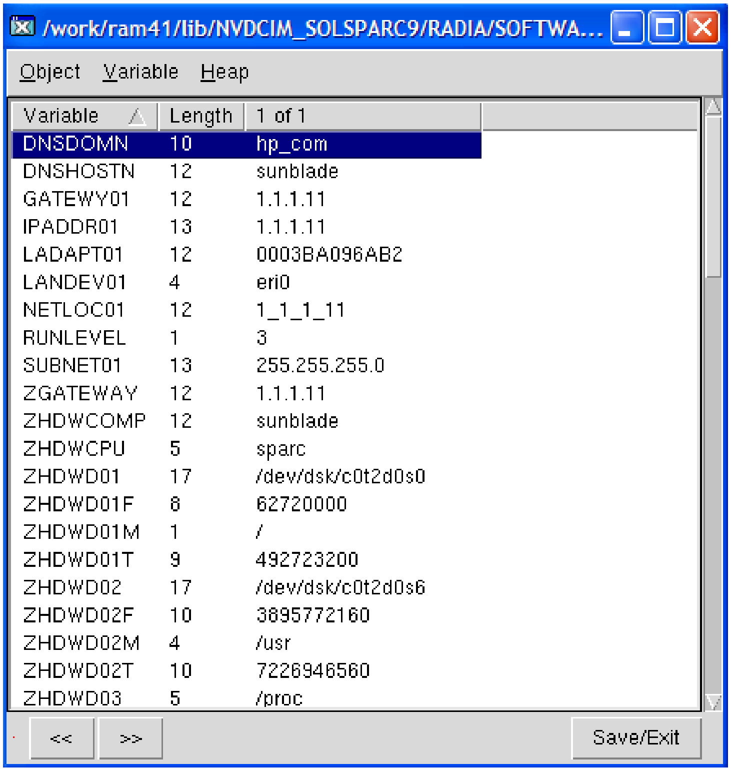
Edit SYSTEM.PROCESS.WBEMAUDT and remove any connection to the SYSTEM.ZMETHOD.POST_WBEM instance.

For more information, refer to the *CM Messaging Server Installation and Configuration Guide*.

Hardware Auditing

Each time an agent connects to the CM 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 CM agent's object directory tree.

Figure 4 ZCONFIG object in CM Admin Agent Explorer

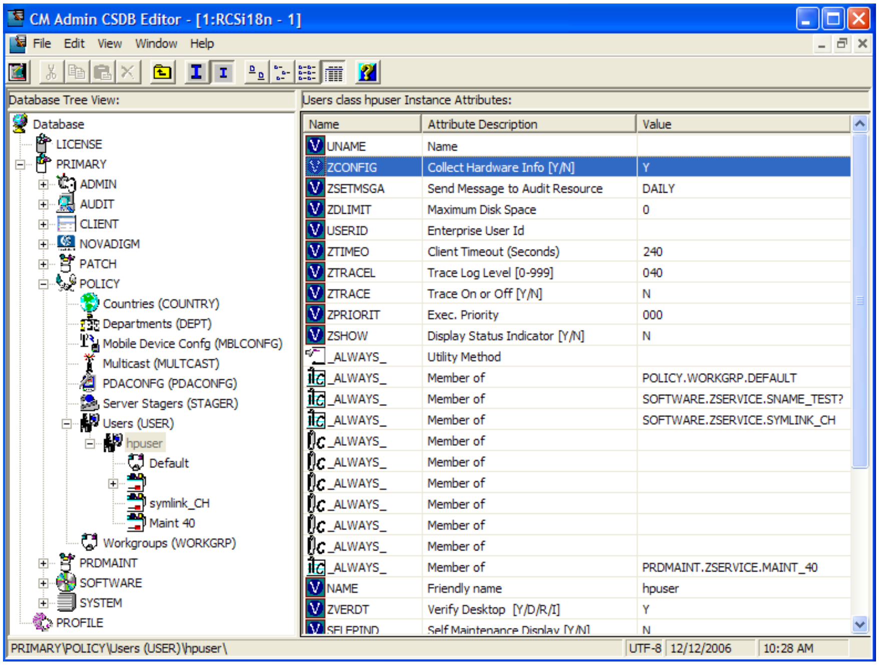


Variable	Length	1 of 1
DNSDOMN	10	hp_com
DNSHOSTN	12	sunblade
GATEWY01	12	1.1.1.11
IPADDR01	13	1.1.1.11
LADAPT01	12	0003BA096AB2
LANDEV01	4	eri0
NETLOC01	12	1_1_1_11
RUNLEVEL	1	3
SUBNET01	13	255.255.255.0
ZGATEWAY	12	1.1.1.11
ZHDWCOMP	12	sunblade
ZHDWCPU	5	sparc
ZHDWD01	17	/dev/dsk/c0t2d0s0
ZHDWD01F	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 CM Admin CSDB Editor, which is available for 32-bit Windows platforms.

Figure 5 POLICY.USER class – ZCONFIG variable



Database Tree View:

- Database
  - LICENSE
  - PRIMARY
    - ADMIN
    - AUDIT
    - CLIENT
    - NOVADIGM
    - PATCH
    - POLICY
      - Countries (COUNTRY)
      - Departments (DEPT)
      - Mobile Device Confg (MBLCONFG)
      - Multicast (MULTCAST)
      - PDACONFG (PDACONFG)
      - Server Stagers (STAGER)
      - Users (USER)
        - hpuser
          - Default
          - symlink_CH
          - Maint 40
      - Workgroups (WORKGRP)
    - PRDMAINT
    - SOFTWARE
    - SYSTEM
  - PROFILE

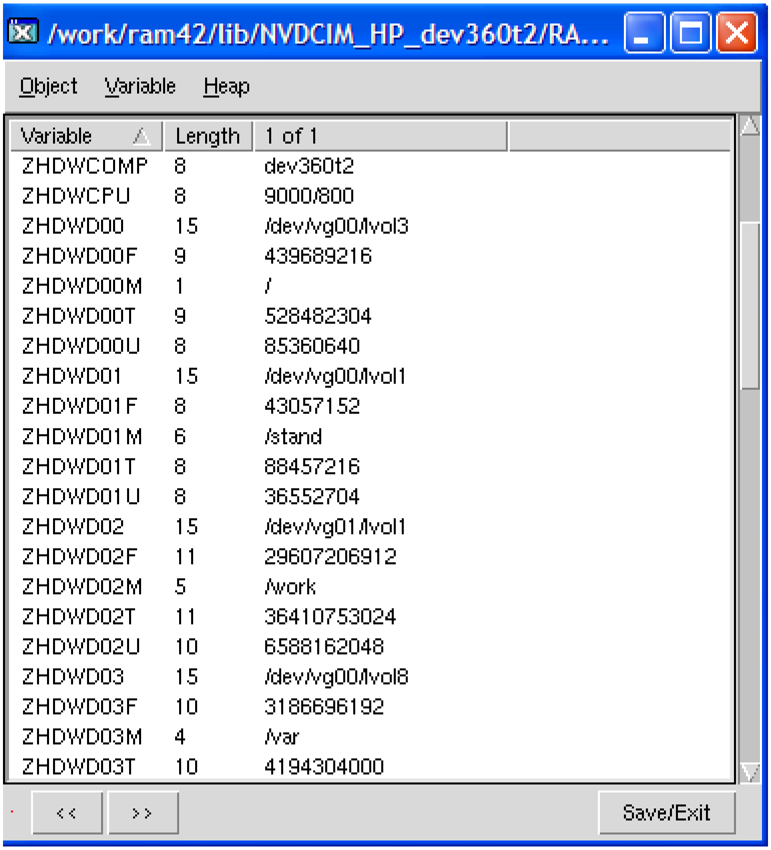
Users class hpuser Instance Attributes:

Name	Attribute Description	Value
UNAME	Name	
ZCONFIG	Collect Hardware Info [Y/N]	Y
ZSETMSGA	Send Message to Audit Resource	DAILY
ZDLIMIT	Maximum Disk Space	0
USERID	Enterprise User Id	
ZTIMEO	Client Timeout (Seconds)	240
ZTRACEL	Trace Log Level [0-999]	040
ZTRACE	Trace On or Off [Y/N]	N
ZPRIORIT	Exec. Priority	000
ZSHOW	Display Status Indicator [Y/N]	N
ALWAYS	Utility Method	
ALWAYS	Member of	POLICY.WORKGRP.DEFAULT
ALWAYS	Member of	SOFTWARE.ZSERVICE.SNAME_TEST?
ALWAYS	Member of	SOFTWARE.ZSERVICE.SYMLINK_CH
ALWAYS	Member of	
ALWAYS	Member of	
ALWAYS	Member of	
ALWAYS	Member of	
ALWAYS	Member of	
ALWAYS	Member of	
ALWAYS	Member of	PRDMAINT.ZSERVICE.MAINT_40
NAME	Friendly name	hpuser
ZVERDT	Verify Desktop [Y/D/R/I]	Y
SELFPIND	Self Maintenance Display [Y/N]	N

PRIMARY\POLICY\Users (USER)\hpuser\ UTF-8 12/12/2006 10:28 AM

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

Figure 6 Sample ZCONFIG object in CM Admin Agent Explorer



Variable	Length	1 of 1
ZHDWCOMP	8	dev360t2
ZHDWCPU	8	9000/800
ZHDWD00	15	/dev/vg00/lvol3
ZHDWD00F	9	439689216
ZHDWD00M	1	/
ZHDWD00T	9	528482304
ZHDWD00U	8	85360640
ZHDWD01	15	/dev/vg00/lvol1
ZHDWD01F	8	43057152
ZHDWD01M	6	/stand
ZHDWD01T	8	88457216
ZHDWD01U	8	36552704
ZHDWD02	15	/dev/vg01/lvol1
ZHDWD02F	11	29607206912
ZHDWD02M	5	/work
ZHDWD02T	11	36410753024
ZHDWD02U	10	6588162048
ZHDWD03	15	/dev/vg00/lvol8
ZHDWD03F	10	3186696192
ZHDWD03M	4	/var
ZHDWD03T	10	4194304000

The ZCONFIG object stores hardware information discovered by the CM 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 19 Attributes in a Sample ZCONFIG

Attribute	Description	Example
DESCRIPT	*Internal use only*	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	HPUX
ZHDWXHID	Host ID	0x1010163

Attribute	Description	Example
ZHDWXHN	Host name	Hpuxdemo
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	royr

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

Summary

- The CM Inventory Manager allows for file, WBEM, and hardware auditing.
- RIMFSCAN and RIMDIFF methods on the agent computer perform the actual file auditing operations by specifying what files to look for.
- The FILEAUDT object contains one heap for each file discovered during the scan for the audit service.
- • The RIMWBEM method constructs a query from the information contained in an instance of the AUDIT.WBEM Class.
- Each time an agent connects to the CM Configuration Server, information about the subscriber's hardware configuration is stored in the ZCONFIG object.
- To force the transfer of the hardware information, the ZCONFIG variable *must* be set to Y in the POLICY.USER Class.
- The ZCONFIG object stores hardware information discovered by the CM agent's standard hardware auditing method.

6 Successful Auditing

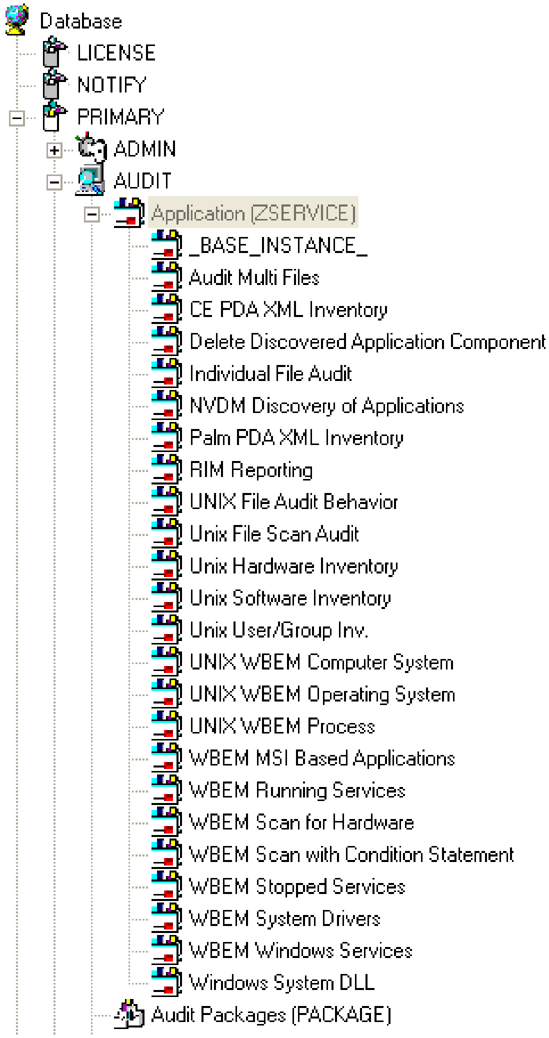
At the end of this chapter, you will:

- 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 CM 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 CM Admin CSDB Editor, which is available for Windows platforms.

Figure 7 Sample Auditing Services in PRIMARY.AUDIT Domain



- Database
  - LICENSE
  - NOTIFY
  - PRIMARY
    - ADMIN
    - AUDIT
      - Application (ZSERVICE)
        - _BASE_INSTANCE_
        - Audit Multi Files
        - CE PDA XML Inventory
        - Delete Discovered Application Component
        - Individual File Audit
        - NVDM Discovery of Applications
        - Palm PDA XML Inventory
        - RIM Reporting
        - UNIX File Audit Behavior
        - Unix File Scan Audit
        - Unix Hardware Inventory
        - Unix Software Inventory
        - Unix User/Group Inv.
        - UNIX WBEM Computer System
        - UNIX WBEM Operating System
        - UNIX WBEM Process
        - WBEM MSI Based Applications
        - WBEM Running Services
        - WBEM Scan for Hardware
        - WBEM Scan with Condition Statement
        - WBEM Stopped Services
        - WBEM System Drivers
        - WBEM Windows Services
        - Windows System DLL
      - Audit Packages (PACKAGE)

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 the following table.

Table 20: 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 a CM 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	This service performs a scan of a systems Win32 devices such as: 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. Note: This is a very large scan and may take several minutes to complete.
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.

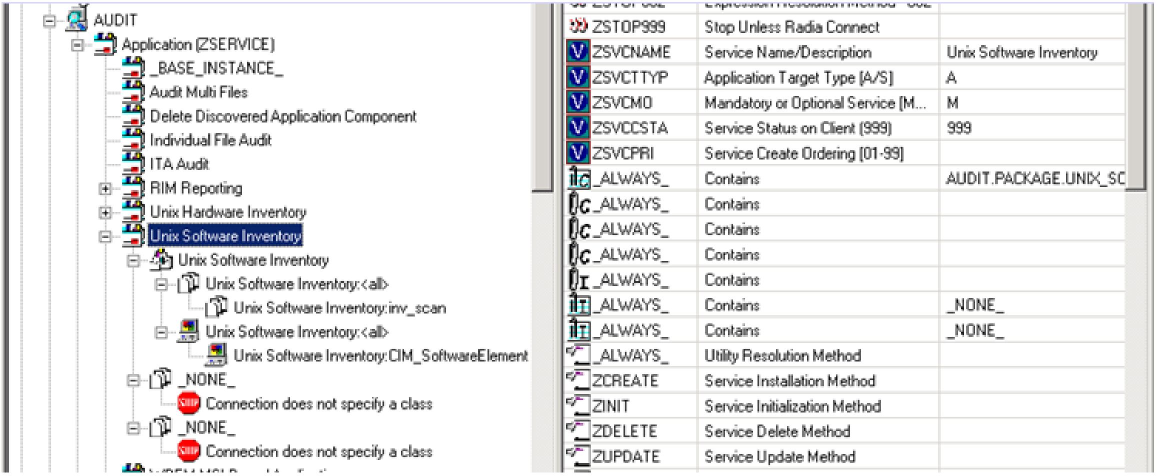
Service	Connected to Audit Package (PACKAGE)	Description
WBEM Running Services	WBEM Scan for Running Services	This service scans the user's computer for system services that are running at the time of the scan.
WBEM Scan for Hardware	WBEM Scan for System Software	This service scans for and reports on 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 *CM Admin CSDB Editor 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 CM Configuration Server for storage.

Figure 8 Unix Software sample audit in AUDIT.ZSERVICE



- AUDIT
  - Application (ZSERVICE)
    - _BASE_INSTANCE_
    - Audit Multi Files
    - Delete Discovered Application Component
    - Individual File Audit
    - ITA Audit
    - RIM Reporting
    - Unix Hardware Inventory
    - Unix Software Inventory
      - Unix Software Inventory
        - Unix Software Inventory:<all>
          - Unix Software Inventory:inv_scan
        - Unix Software Inventory:<all>
          - Unix Software Inventory:CIM_SoftwareElement
      - _NONE_
        - Connection does not specify a class
      - _NONE_
        - Connection does not specify a class

ZSTOP999	Stop Unless Radia Connect	
ZSVCNAME	Service Name/Description	Unix Software Inventory
ZSVCTTYP	Application Target Type [A/S]	A
ZSVCMO	Mandatory or Optional Service [M...	M
ZSVCCSTA	Service Status on Client (999)	999
ZSVCPRI	Service Create Ordering [01-99]	
ALWAYS	Contains	AUDIT.PACKAGE.UNIX_SC
ALWAYS	Contains	
ALWAYS	Contains	
ALWAYS	Contains	
ALWAYS	Contains	
ALWAYS	Contains	_NONE_
ALWAYS	Contains	_NONE_
ALWAYS	Utility Resolution Method	
ZCREATE	Service Installation Method	
ZINIT	Service Initialization Method	
ZDELETE	Service Delete Method	
ZUPDATE	Service Update Method	

To configure a sample Audit package

1 If you have not already done so, start the CM 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.

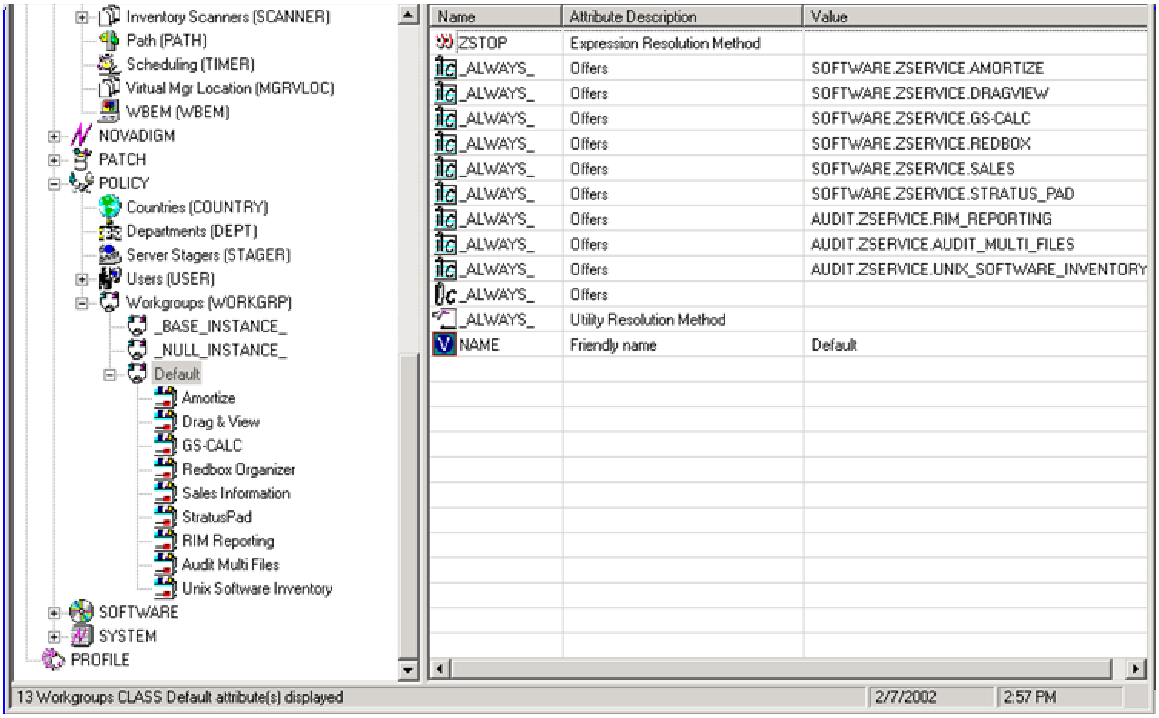
The Select Connection Attribute window opens.

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.



- Inventory Scanners (SCANNER)
- Path (PATH)
- Scheduling (TIMER)
- Virtual Mgr Location (MGRVLOC)
- WBEM (WBEM)
- NOVADIGM
- PATCH
- POLICY
  - Countries (COUNTRY)
  - Departments (DEPT)
  - Server Stagers (STAGER)
  - Users (USER)
  - Workgroups (WORKGRP)
    - _BASE_INSTANCE_
    - _NULL_INSTANCE_
    - Default
      - Amortize
      - Drag & View
      - GS-CALC
      - Redbox Organizer
      - Sales Information
      - StratusPad
      - RIM Reporting
      - Audit Multi Files
      - Unix Software Inventory
- SOFTWARE
- SYSTEM
- PROFILE

Name	Attribute Description	Value
ZSTOP	Expression Resolution Method	
ALWAYS	Offers	SOFTWARE.ZSERVICE.AMORTIZE
ALWAYS	Offers	SOFTWARE.ZSERVICE.DRAGVIEW
ALWAYS	Offers	SOFTWARE.ZSERVICE.GS-CALC
ALWAYS	Offers	SOFTWARE.ZSERVICE.REDBOX
ALWAYS	Offers	SOFTWARE.ZSERVICE.SALES
ALWAYS	Offers	SOFTWARE.ZSERVICE.STRATUS_PAD
ALWAYS	Offers	AUDIT.ZSERVICE.RIM_REPORTING
ALWAYS	Offers	AUDIT.ZSERVICE.AUDIT_MULTI_FILES
ALWAYS	Offers	AUDIT.ZSERVICE.UNIX_SOFTWARE_INVENTORY
ALWAYS	Offers	
ALWAYS	Utility Resolution Method	
NAME	Friendly name	Default

13 Workgroups CLASS Default attribute(s) displayed 2/7/2002 2:57 PM

The collection of inventory information occurs on the CM agent computer when a user connects to the CM Configuration Server through the CM 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

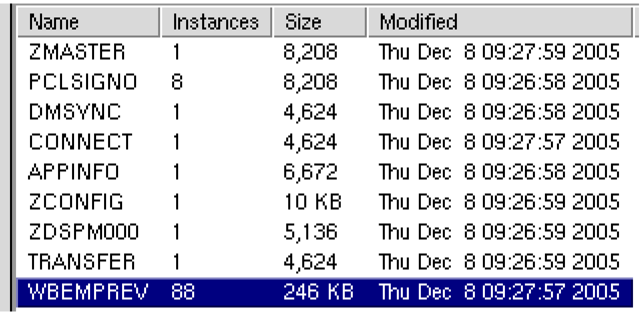
Use the CM Admin Agent Explorer to locate the ZSERVICE instance for the Unix Software Inventory package in the `LIB` directory.

To locate the ZSERVICE object using the CM Admin Agent Explorer

1 Start the CM 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/SOFTWWARE/ZSERVICE/UNIX_SO
FTWARE_INVENTORY
```



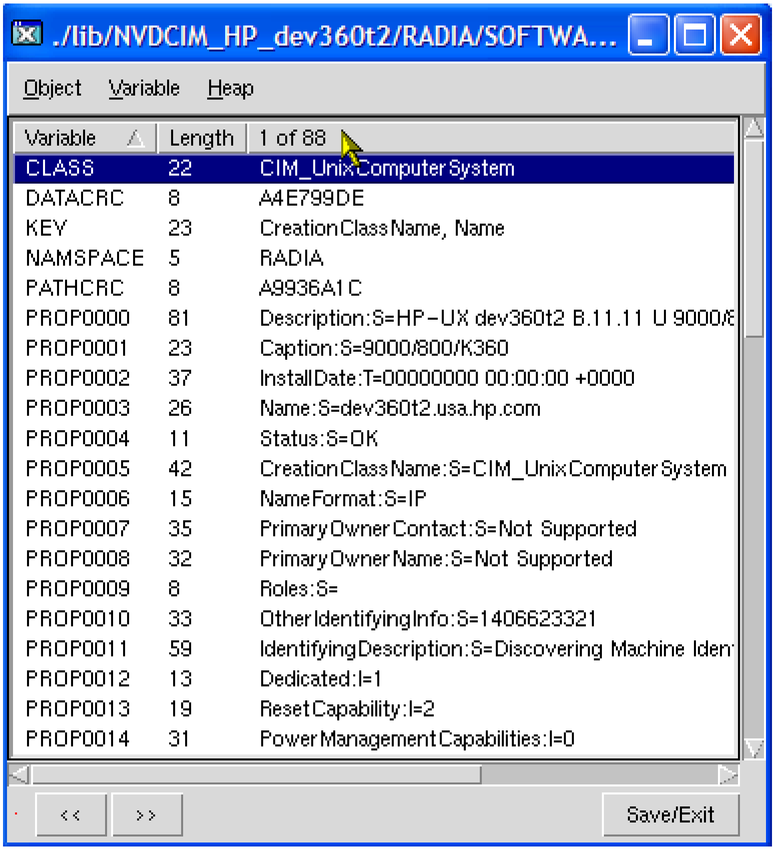
Name	Instances	Size	Modified
ZMASTER	1	8,208	Thu Dec 8 09:27:59 2005
PCLSIGNO	8	8,208	Thu Dec 8 09:26:58 2005
DMSYNC	1	4,624	Thu Dec 8 09:26:58 2005
CONNECT	1	4,624	Thu Dec 8 09:27:57 2005
APPINFO	1	6,672	Thu Dec 8 09:26:58 2005
ZCONFIG	1	10 KB	Thu Dec 8 09:26:59 2005
ZDSPM000	1	5,136	Thu Dec 8 09:26:59 2005
TRANSFER	1	4,624	Thu Dec 8 09:26:59 2005
WBEMPREV	88	246 KB	Thu Dec 8 09:27:57 2005

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 CM 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 CM 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 CM Configuration Server.
- The CM agent then deletes the WBEMAUDT object and will rename the WBEMCURR object to WBEMPREV.

Figure 9 WBEMPREV heaps in CM Admin Agent Explorer



Variable	Length	1 of 88
CLASS	22	CIM_UnixComputerSystem
DATACRC	8	A4E799DE
KEY	23	CreationClassName, Name
NAMSPACE	5	RADIA
PATHCRC	8	A9936A1C
PROP0000	81	Description:S=HP-UX dev360t2 B.11.11 U 9000/8
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 Iden
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.

Summary

- To illustrate the concepts of inventory information collection, the CM Inventory Manager installation contains a set of representative audit service examples.
- The best way to develop your own audit services is to study the samples that were installed with the CM Inventory Manager.
- The collection of inventory information occurs on the CM agent computer when a user connects to the CM Configuration Server.
- The first connection downloads the audit service. The second connection sends the audit results back to the CM Configuration Server. The audit-related scans are done between the two connections.

7 Creating Audit Packages

At the end of this chapter, you will:

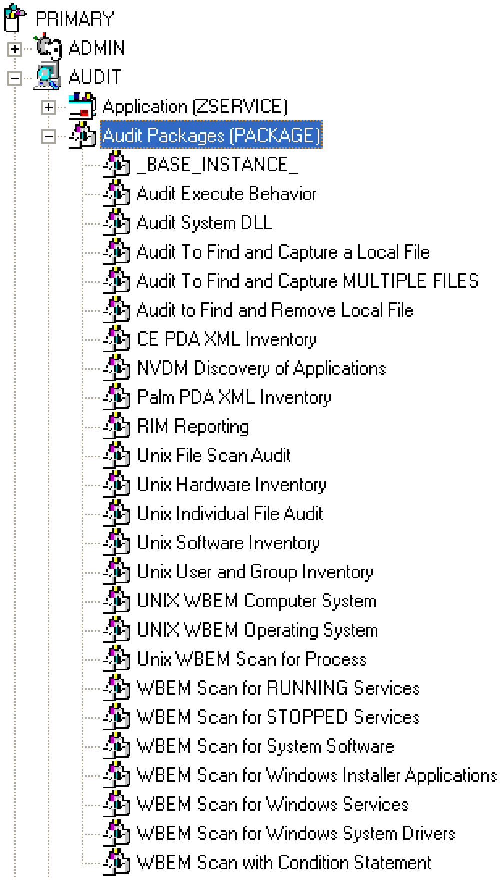
- Have created a new file audit package.
- Have created a new ZSERVICE for your package.

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

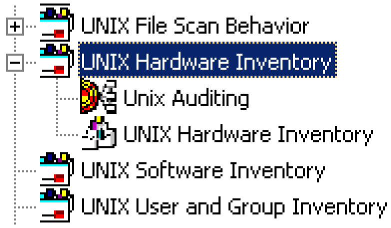


- PRIMARY
  - ADMIN
  - AUDIT
    - Application (ZSERVICE)
    - Audit Packages (PACKAGE)
      - _BASE_INSTANCE_
      - Audit Execute Behavior
      - Audit System DLL
      - Audit To Find and Capture a Local File
      - Audit To Find and Capture MULTIPLE FILES
      - Audit to Find and Remove Local File
      - CE PDA XML Inventory
      - NVDM Discovery of Applications
      - Palm PDA XML Inventory
      - RIM Reporting
      - Unix File Scan Audit
      - Unix Hardware Inventory
      - Unix Individual File Audit
      - Unix Software Inventory
      - Unix User and Group Inventory
      - UNIX WBEM Computer System
      - UNIX WBEM Operating System
      - Unix WBEM Scan for Process
      - WBEM Scan for RUNNING Services
      - WBEM Scan for STOPPED Services
      - WBEM Scan for System Software
      - WBEM Scan for Windows Installer Applications
      - WBEM Scan for Windows Services
      - WBEM Scan for Windows System Drivers
      - WBEM Scan with Condition Statement

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



UNIX File Scan Behavior
UNIX Hardware Inventory
Unix Auditing
UNIX Hardware Inventory
UNIX Software Inventory
UNIX User and Group Inventory

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 21 Inventory Options (RIMOPTS) Class

Instance	Description
Default	Contains the base instance attributes for the RIMOPTS Class. • 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. • 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	When connected to an audit service, will difference the audited information in initial installation, on subsequent connects, and when updated. • Collect attribute is set to Diff. • Runexec attribute is set to IVU. • ZSVCTYPE attribute is set to I.
Full Audit on Install and Update	When connected to an audit service, will difference the audited information on installation and update. • 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	When connected to an audit service, will • Collect attribute is set to Full. • Runexec attribute is set to IVU. • ZSVCTYPE attribute is set to I.

Using CM Admin CSDB Editor to Create and Maintain Audit Services

We will use the CM 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 CM Admin CSDB Editor is available for Windows platforms. For more information, refer to the *CM Admin CSDB Editor 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 OVCM Administrator** → **CM Admin 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 CM 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 CM Admin CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Expand the AUDIT Domain.
5. Double-click the **Audit Packages (PACKAGE) Class**.

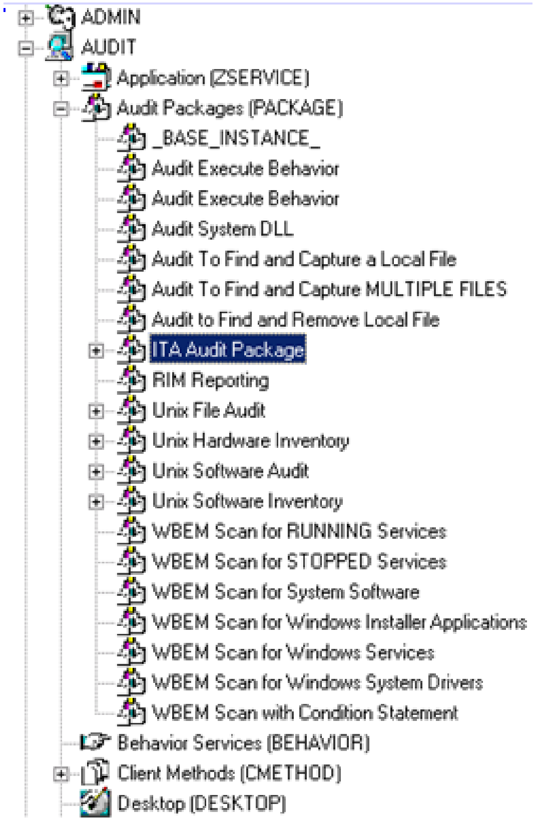
   As an example, we will create a new auditing package called ITA Audit Package. This package will scan a user's computer, capture logical disk information, and return the results to the administrator.

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 CM 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.


ADMIN
AUDIT
Application (ZSERVICE)
Audit Packages (PACKAGE)
_BASE_INSTANCE_
Audit Execute Behavior
Audit Execute Behavior
Audit System DLL
Audit To Find and Capture a Local File
Audit To Find and Capture MULTIPLE FILES
Audit to Find and Remove Local File
ITA Audit Package
RIM Reporting
Unix File Audit
Unix Hardware Inventory
Unix Software Audit
Unix Software Inventory
WBEM Scan for RUNNING Services
WBEM Scan for STOPPED Services
WBEM Scan for System Software
WBEM Scan for Windows Installer Applications
WBEM Scan for Windows Services
WBEM Scan for Windows System Drivers
WBEM Scan with Condition Statement
Behavior Services (BEHAVIOR)
Client Methods (CMETHOD)
Desktop (DESKTOP)

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

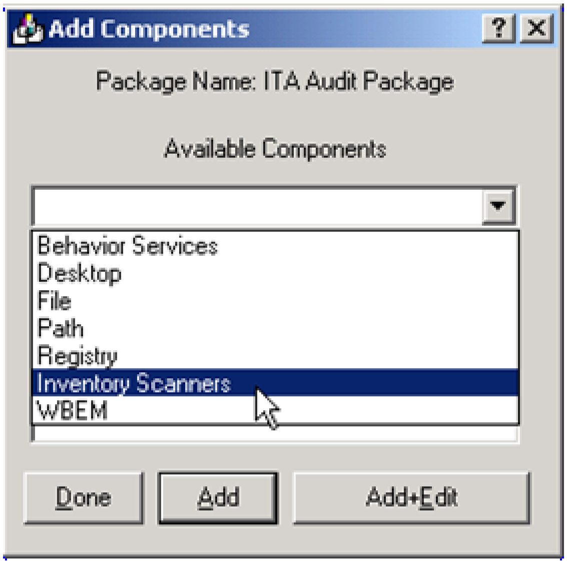
To add a component to an Audit package

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 **`nvdcim`**. This is the name of the Tcl script that will be executed by the client to initiate the inventory scan. The nvdcim 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.

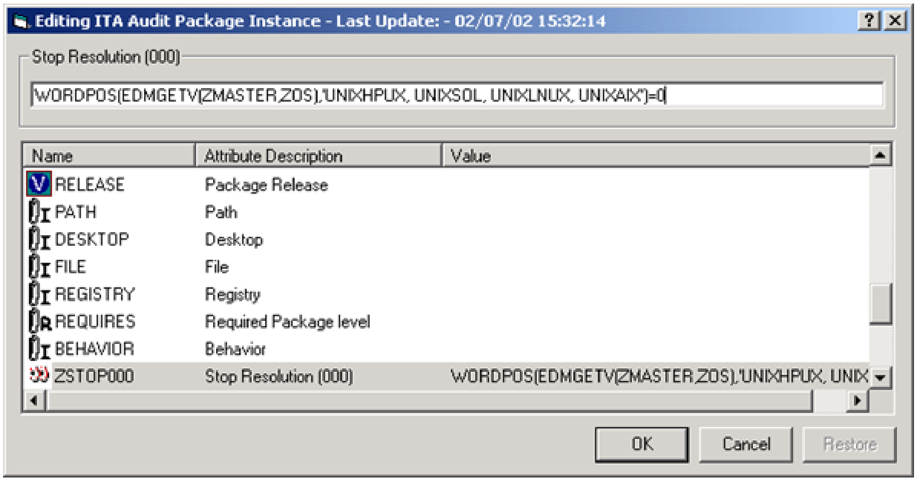
Now edit the package class instance ZSTOP expression to reflect the supported UNIX platforms. The default ZSTOP expression is configured for Windows platforms.

To update the ZSTOP expression

1 In the tree view of the CM Admin CSDB Editor, double-click the new audit package name, **ITA Audit Package**.

2 In the list view of the CM Admin CSDB Editor, double-click the **ZSTOP** expression.

3 Replace the supported Windows platform names with the appropriate UNIX platforms.



Editing ITA Audit Package Instance - Last Update: - 02/07/02 15:32:14

Stop Resolution (000)

WORDPOS(EDMGETV(ZMASTER,ZOS),'UNIXHPUX, UNIXSOL, UNIXLNUX, UNIXAIX')=0

Name	Attribute Description	Value
RELEASE	Package Release	
PATH	Path	
DESKTOP	Desktop	
FILE	File	
REGISTRY	Registry	
REQUIRES	Required Package level	
BEHAVIOR	Behavior	
ZSTOP000	Stop Resolution (000)	WORDPOS(EDMGETV(ZMASTER,ZOS),'UNIXHPUX, UNIX

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 CM 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 CM 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 OVCM Administrator** → **CM Admin 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 CM 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 CM Admin CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Expand the **AUDIT Domain**.
5. 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.
6. Right-click the **Audit Packages (PACKAGE) Class**.
7. Select **New Instance** from the menu.
8. Type a new display name for the package instance. This is the friendly name that will appear in the tree view.
9. 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 CM Admin CSDB Editor window when the instance is selected and opened in the tree view.
10. Click **OK** to continue.

    The new Audit Package is added to the AUDIT.PACKAGE Class.
11. 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 96 for instructions on creating a ZSERVICE and removing the required ALWAYS connections.

Summary

- 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. The fact that a package instance owns a component class instance means that all of the instances are managed as a package unit.
- By viewing and deploying the sample audits provided by HP, systems administrators will be able to create and use their own auditing packages.
- The New Application Wizard is *not* available to connect a package to a service within the Audit domain. You need to either copy an existing instance or create a new one.

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 CM 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.

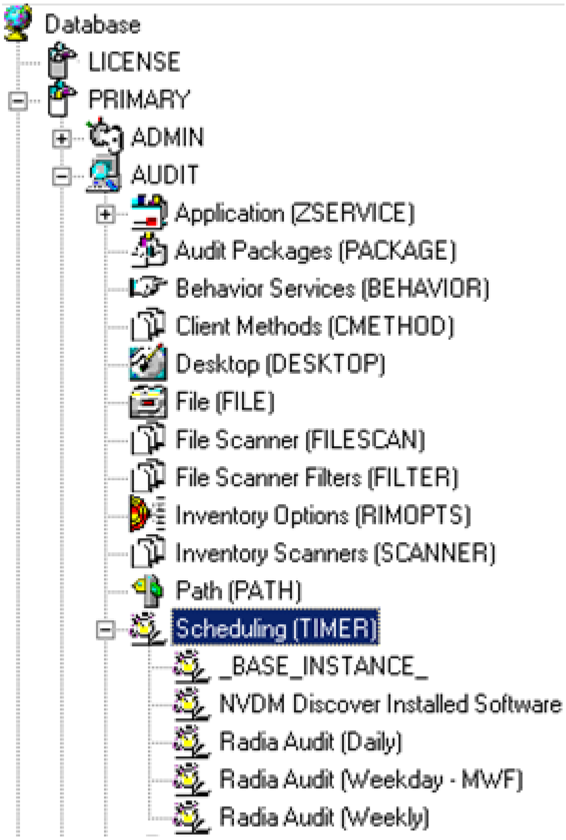
> As distributed by HP, the SOFTWARE Domain also contains a Scheduling (TIMER) Class. Timers can be specified in instances of either Scheduling (TIMER) Class and can be connected to an Application (ZSERVICE) Class instance in either the SOFTWARE or AUDIT Domains interchangeably.

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 *CM Admin CSDB Editor 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 107.

Figure 12 AUDIT Scheduling (TIMER) Class



- Database
  - LICENSE
  - PRIMARY
    - ADMIN
    - AUDIT
      - Application (ZSERVICE)
      - Audit Packages (PACKAGE)
      - Behavior Services (BEHAVIOR)
      - Client Methods (CMETHOD)
      - Desktop (DESKTOP)
      - File (FILE)
      - File Scanner (FILESCAN)
      - File Scanner Filters (FILTER)
      - Inventory Options (RIMOPTS)
      - Inventory Scanners (SCANNER)
      - Path (PATH)
      - Scheduling (TIMER)
        - _BASE_INSTANCE_
        - NVDM Discover Installed Software
        - Radia Audit (Daily)
        - Radia Audit (Weekday - MWF)
        - Radia Audit (Weekly)

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 CM agent is installed with the CM Scheduler service. This service contains an executable timer component that executes any program on the end-user desktop when a timer expires.

Typically, the CM 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 CM Configuration Server to deploy or maintain a service.

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

Table 22 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 *before* 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	*Used only when* ZSCHFREQ = PERIODIC. Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4 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. Example 1: Suppose you create and deploy a timer with the ZSCHDEF = DAILY(&ZSYSDATE,04:00:00) If ZSCHTYPE = IMMEDIATE and it is: • 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 If ZSCHTYPE = DEFERRED and it is: • Before 04:00:00, the command in the instance will be executed the *next* day at 04:00:00 • After 04:00:00, the command in the instance will be

Attribute	Description
	executed the *next* day at 04:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,04:00:00) If ZSCHTYPE = IMMEDIATE and it is: • 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 If ZSCHTYPE = DEFERRED and it is: • 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	Indicates how often the timer should expire according to the frequency specified in the ZSCHDEF attribute. • Once for a one-time expiration. • Periodic for a repeated expiration. • Random for random intervals.
ZRSCCMDL	Indicates the command line that is executed on the subscriber's computer when the timer expires.
ZSVCOID	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.
ALWAYS	Stores the connections to other instances.
NAME	The friendly name for this instance.
APPSVC	The Application Name.
REQUEST	The Application Request.
DOMAIN	The server's domain name.
IPADDR	The server's IP address/name.
SOCKET	The server's socket number.

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 CM Configuration Server. An expired time will continually evaluate whether communications with the CM 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 CM 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 CM 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 CM *Application Manager Guide*.

The following section uses the CM 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** → **Administrator Workstation** → **CM Admin CSDB Editor**.

The CM 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 CM security administrator to obtain your own User ID and Password, if necessary.

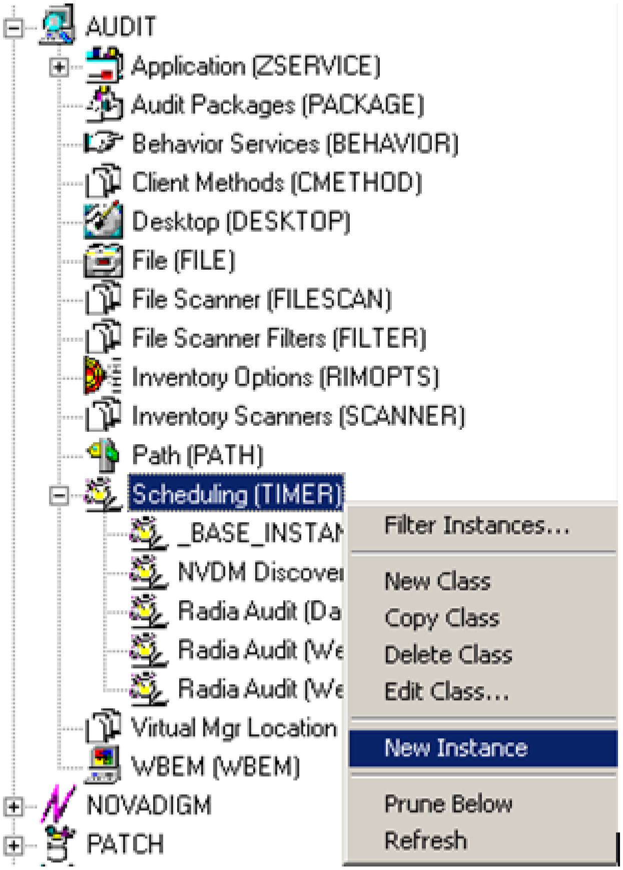
2 If necessary, type a User ID and Password, and then click **OK**. The CM Admin CSDB Editor window opens.

3 Double-click **PRIMARY**.

4 Expand the **AUDIT Domain**.

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

A shortcut menu opens.

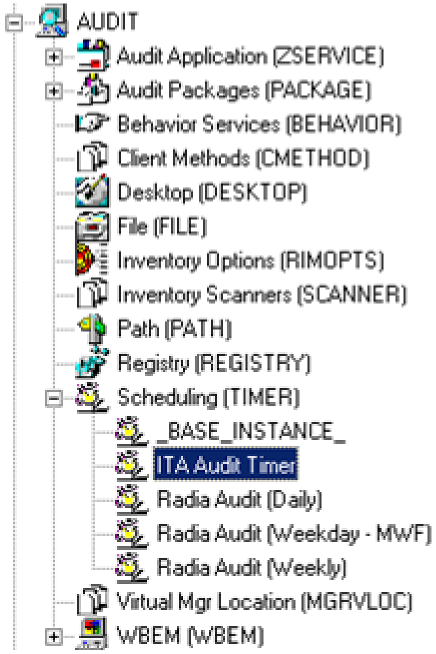


6 Select **New Instance**.

The Create Instance dialog box opens.

7 Type a name for the new timer instance.

8 Click **OK**. The 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 CM *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 the following table to help you determine the appropriate syntax.

Table 23 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.
WEEKDAY	WEEKDAY(MONDAY,01:00:00)	Every *Name of Weekday** 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 CM Configuration Server. Use this attribute when there is a possibility that the client will not be able to connect with the CM 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 CM Configuration Server.
- If ZNOPING = N, the Scheduler service will ping the CM Configuration Server.
    - If the CM 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 CM Configuration Server again.
    - If the CM 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 CM 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 CM agent connects to the CM Configuration Server.

Use the CM 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 CM 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 CM

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 CM 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.

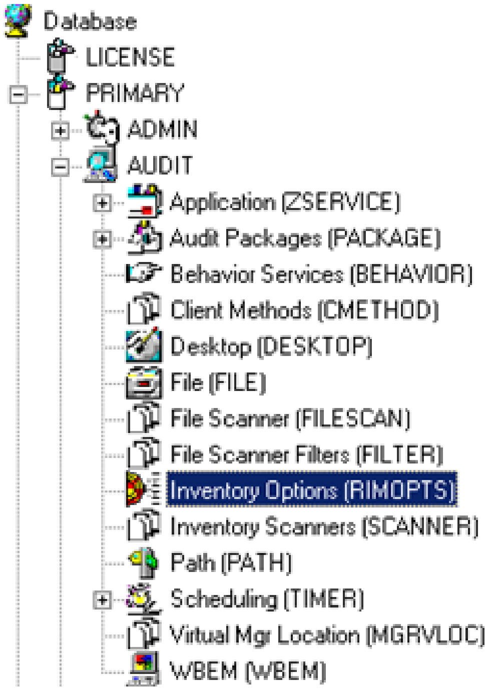
To customize the RIMOPTS instance

1 From the **Start** menu, select **Programs** → **HP OVCM Administrator** → **CM Admin 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 CM 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.

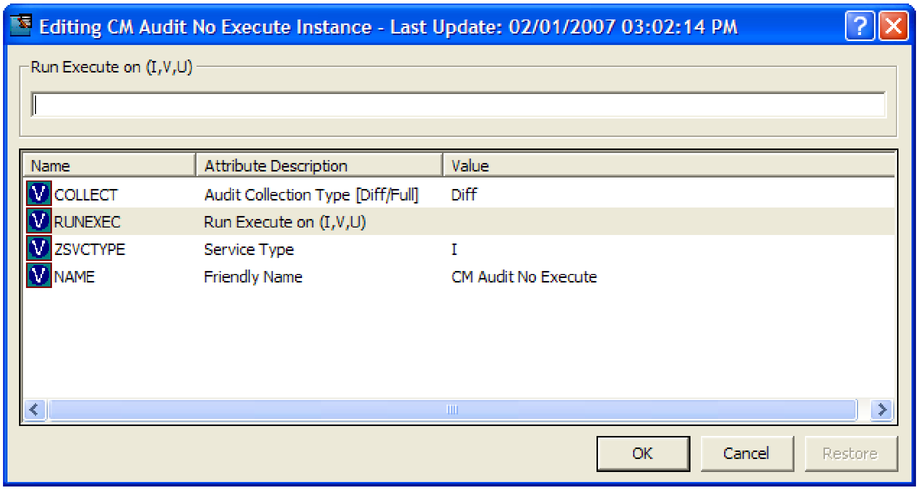
3 Expand the **PRIMARY File** and the **AUDIT Domain**.



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.



Editing CM Audit No Execute Instance - Last Update: 02/01/2007 03:02:14 PM

Run Execute on (I,V,U)

Name	Attribute Description	Value
COLLECT	Audit Collection Type [Diff/Full]	Diff
RUNEXEC	Run Execute on (I,V,U)	
ZSVCTYPE	Service Type	I
NAME	Friendly Name	CM Audit No Execute

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

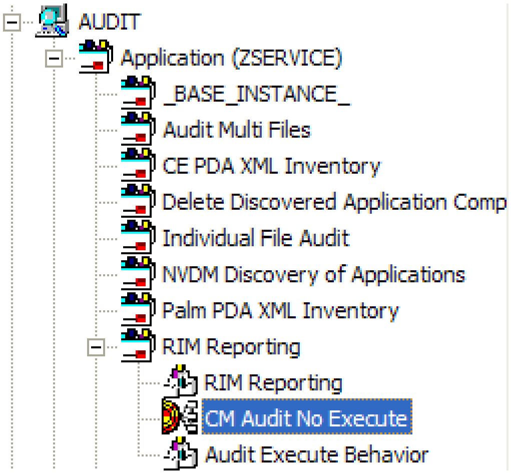
7 Right-click the **RIM_REPORTING** Service in the AUDIT Class.

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 CM agent.

The CM Audit No Execute instance is now connected to the RIM Reporting service.



- AUDIT
  - Application (ZSERVICE)
    - _BASE_INSTANCE_
    - Audit Multi Files
    - CE PDA XML Inventory
    - Delete Discovered Application Comp
    - Individual File Audit
    - NVDM Discovery of Applications
    - Palm PDA XML Inventory
    - RIM Reporting
      - RIM Reporting
      - CM Audit No Execute
      - Audit Execute Behavior

Summary

- The Scheduling (TIMER) Class enables the CM administrator to set a timer on the agent computer and will cause one or more audit services to be processed whenever the timer expires.
- 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.
- Typically, the CM Scheduler service lies dormant in the background, and wakes up once per minute to see if a timer has expired.
- Use ZSCHDEF to indicate when the timer should expire.
- Use ZRSCCMDL to execute a command on the subscriber's computer when the timer expires.

9 Viewing Inventory from the CM Reporting Server

At the end of this chapter, you will:

- Know how to access and use the CM 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 CM Reporting Server

To access the CM 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 CM Reporting Server was installed and where `reportingserver` is the alias assigned to CM Reporting Server during its installation.

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

Viewing Audit Information Using the CM Reporting Server

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

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

Reporting Views for Inventory Reports

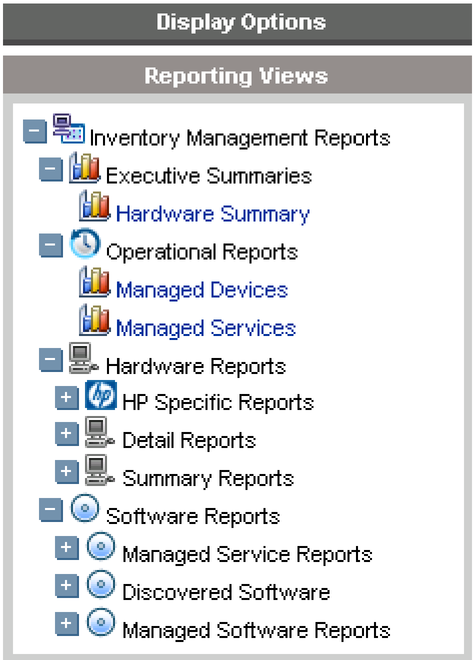
To view the reports, first access your CM Reporting Server. Then, under Reporting Views, click **Inventory Management Reports** to expand the list of reports.

There are different types of inventory reports:

- Executive Summaries
- Operational Reports
- Hardware Reports

- Software Reports.

Figure 13 Inventory Management Reports



Display Options

Reporting Views

- Inventory Management Reports
  - Executive Summaries
    - Hardware Summary
  - Operational Reports
    - Managed Devices
    - Managed Services
  - Hardware Reports
    - HP Specific Reports
    - Detail Reports
    - Summary Reports
  - Software Reports
    - Managed Service Reports
    - Discovered Software
    - Managed Software Reports

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

Table 24 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 25 Software Reporting Views

Reporting View Types	Reporting Views
Managed Service Reports	Service Summary Service Details
Discovered Software	Vendor Reports • Discovered Software by Vendor Product Reports • Discovered Software by Product • Discovered Software by Version Application Reports • Discovered Software by Application • Discovered Software by Application Version
Managed Software Reports	Vendor Reports • Managed Software by Vendor Product Reports • Managed Software by Product • Managed Software by Product Version Application Reports • Managed Software by

Reporting View Types	Reporting Views
	Application • Managed Software by Application Version

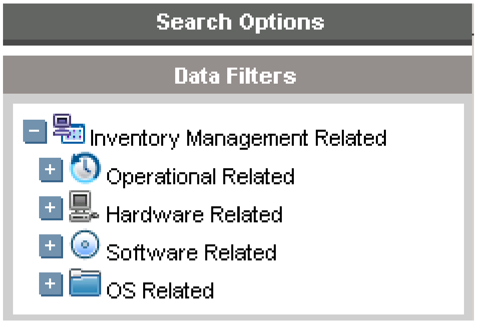
Filtering Inventory Reports with CM Reporting Server

CM Reporting Server provides extensive filtering capabilities. To access the filters, expand Inventory Manager Related in the Search Controls section of the CM 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 CM Reporting Server in general, refer to the *CM Reporting Server Guide*.

Summary

- Use a web browser to access the CM Reporting Server to view reports on collected hardware, software, and operational information.
- Select an Inventory Management Reports Reporting View to display collected data.
- Apply Inventory Management Related Data Filters to modify the data displayed in the current Reporting View.

A Product Name Changes

If you have used Radia in the past, and are not yet familiar with the newly rebranded HP terms and product names, Table 26 below will help you identify naming changes that have been applied to the Radia brand.

Table 26 Product Name and Term Changes

New Name/Term	Old Name/Term
CM A gent Installation Wizard	Radia Client Installation Wizard
CM agents	Radia clients
HP OpenView Configuration Management Administrator	Radia Administrator Workstation
HP OpenView Configuration Management	Radia
HP OpenView Configuration Management Admin Agent Explorer	Radia Client Explorer
HP OpenView Configuration Management Admin CSDB Editor	Radia System Explorer
HP OpenView Configuration Management Admin Packager	Radia Packager
HP OpenView Configuration Management Admin Screen Painter	Radia Screen Painter
HP OpenView Configuration Management Application Usage Manager	Radia Usage Manager
HP OpenView Configuration Management Solutions for Servers	Server Management

Index