HP Configuration Management

Inventory Manager

for the Windows® operating system

Software Version: 5.10

Installation and Configuration Guide

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

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

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

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

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

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

OpenSSL License
Copyright © 1998-2001 The OpenSSLProject.

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

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

Documentation Updates

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:

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

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

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

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 5.10 release.

Table 2 lists features added for the Configuration Management v 5.00 release.

Table 1   New features added for Configuration Management 5.10

Chapter	Version	Changes
	5.10	The CM Administrator installation chapter was removed and is now included in the CM Administrator User Guide.
3	5.10	Page 35, About the Installation Process: CM-MgmtApps50.msi has been renamed to CM-MgmtApps50.msi.
4	All	Page 79, Implementing Registry Scans, topic modified to use the Add Component dialog to both Add and Edit a registry scan instance from an AUDIT.PACKAGE instance.
4	All	Page 76, RIMOPTS Class, corrected the procedure: To enable drag-and-drop connections for RIMOPTS Class instances. Changed the CSDB class needed to access the CONNECT_ZSERVICE_TO_RULES instance from: **PRIMARY → ADMIN → Name Lists (32) (ZLIST32)** to **PRIMARY → ADMIN → Name Lists (8) (ZLIST).**

Chapter	Version	Changes
8	all	Page 138, Table 21, Scheduling Timer Class, ZSCHTYPE Row, timer examples now show the required leading zero to indicate 4 AM: e.g., ZSCHDEF = DAILY(&ZSYSDATE,**04**:00:00).

Table 2   New features added for Configuration Management 5.00

Chapter	Version	Changes
3	5.00	Page 25, 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 34, System Requirements for installing the agents have changed.
3	5.00	Page 35, About the Installation Process: Radia401.msi has been renamed to CM-MgmtApps50.msi.
5	5.00	Page 99, WBEM Object Processing, discusses how WBEM objects are collected and posted by the WBEM Data Delivery Agent of the CM Messaging Server to the Inventory ODBC database.
		Replaces the previous topic, WBEM Objects and the CM Configuration Server.
9	5.00	Page 155, Viewing Inventory from the CM Reporting Server, new chapter. Replaces the earlier chapter: "Viewing Inventory from the Radia Integration Server".
10	5.00	Page 159, Windows Vista Readiness Reports, new reports are available from the Inventory Reports within the CM Reporting Server.

Table 3 indicates changes made to this document for earlier releases.

Table 3   Document changes

Chapter	Version	Changes
1	4.0	Page 14, About the CM Inventory Manager: revised the text to include the role of the HP- CM Messaging Server Using Radia (CM Messaging Server). You can use the CM Messaging Server instead of the Inventory Manager Server to post discovered inventory data to an ODBC-compliant database.
1	4.1	Page 14, Overview: added the following note regarding the role of the new CM Messaging Server: "As of Radia 4.1, the CM Messaging Server handles the actual delivery of the inspection results to the Radia Integration Server (or directly to the back-end Inventory Manager Database), which frees up CM Configuration Server resources. See the *CM Messaging Server Guide* for more information."
3	4.1	Page 35, About the Installation Process: The sample install.ini is now in the \win32 directory and not the \win32\samples directory.
3	4.0.1	Page 35, About the Installation Process: Radia40.msi has been renamed to Radia401.msi.
3	4.0	Page 40, [PROPERTIES] Section of INSTALL.INI: The following have been added to the properties section of the install.ini file: NVDSTARTWMICFGMGR, NVDRAMSHORTCUT, NVDRAMSTARTMENUSHORTCUT; NVDRAMCONNECT, NVDMAINTDIR, NVDNOTIFYINTERACT, NVDREDIRECTORINTERACT, NVDSCHEDULERINTERACT.
3	4.0.1	Page 40, [PROPERTIES] Section of INSTALL.INI: NVDINSTALLSVR has been added as a value of ADDLOCAL. This allows you to install the Radia Server Management Agent. Only install the CM agents for which you are licensed.
4	4.1	Page 74, Table 14 AUDIT.FILE Class Instances: Added clarification that FILESCAN and FILTER classes are for UNIX devices, only.

Chapter	Version	Changes
5	4.1	Page 92, Table 14 AUDIT.FILE Class Instances: ZMD5INFO row added. The ZMD5INFO attribute permits the collection of MD5 information for a file.
5	4.1	Page 93, Table 15: Added a description for the RESOLVE attribute.

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Contents

1 Introduction

At the end of this chapter, you will:

- Understand the components of the HP Configuration Management Inventory Manager (CM Inventory Manager) agent.

- Understand the terminology associated with the CM Inventory Manager agent.

- Be familiar with supported platforms.

- Be familiar with WBEM and the HP Configuration Management (CM) agent.

- Be familiar with the HP OVCM Integration Server.

- Be familiar with related CM infrastructure components for processing inventory agent data, such as the HP Configuration Management CM Messaging Server (CM Messaging Server), the HP Configuration Management Reporting Server (CM Reporting Server), and the HP Configuration Management Portal (CM Portal).

About the CM Inventory Manager

The CM Inventory Manager agent is a utility used to discover configuration information on remote computers. It enables centralized reporting and administration based upon the discovery results.

Use the CM Inventory Manager agent with the CM Messaging Server to maintain the discovery information within an ODBC-compliant (Open Database Connectivity) database.

Use the CM Reporting Server to view the data reported from the CM Inventory Manager.

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

- For more information on using the CM Messaging Server to create the SQL tables for the CM Inventory ODBC database and post data to your inventory database, refer to the *CM Messaging Server Installation and Configuration Guide (CM Messaging Server Guide)*. The Data Delivery Agents for CORE, INVENTORY and WBEM objects provide this capability.

- For more information on using the CM Reporting Server for creating and obtaining reports, refer to the *CM Reporting Server Guide.*

Overview

Systems administrators use the HP Configuration Management Administrator Configuration Server Database Editor (CM Admin CSDB Editor) or the CM Portal to manipulate the contents of the CM Configuration Server Database (CM-CSDB). They specify what inventory management tasks to perform and on which agent computers to perform them.

The collection of inventory information is performed on the CM Inventory Manager computer when a subscriber connects to and installs the auditing software with the CM Configuration Server as follows:

- **CM Application Self-service Manager** installs the software when the user selects an application to be installed.

- **CM Application Manager** installs the software in one of the following ways:

— through a logon script.

— when the user double-clicks a desktop HP Connect icon.

— according to a schedule.

— using the Notify capabilities of the CM Portal or the CM Configuration Server.

The results of the inspection are then sent back to the CM Configuration Server. Any unwanted files discovered on a user's computer can be captured and/or deleted. This enables administrators to remove unauthorized content, such as games, from end users' computers.

The CM Configuration Server can store these inspection results in its Database PROFILE File and/or have the CM Messaging Server route them for posting to an SQL-compliant database using ODBC.

> By default, the Data Directory Agents of the CM Messaging Server post the inspection results directly to the ODBC database.

The CM Messaging Server handles the actual delivery of the inspection results directly to the CM Inventory Manager database, or, to another CM Messaging Server. Both of these options free up CM Configuration Server resources. Refer to the *CM Messaging Server Guide* for more information.

About the CM Reporting Server

As part of the extended infrastructure for CM, the web-based CM Reporting Server allows you to query the combined data in existing CM Inventory Manager, CM Patch Manager, and CM Application Usage Manager Databases and obtain Executive, Summary, and 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 environment is illustrated in the figure below.

Figure 1   Reporting environment
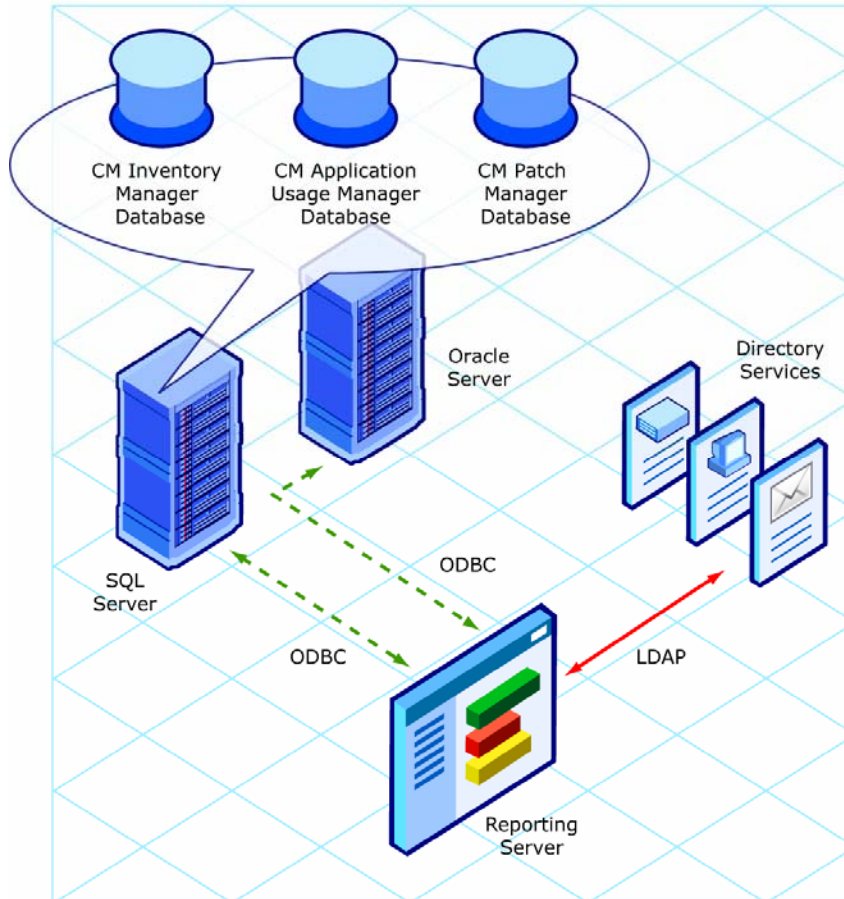
The CM Reporting Server interface provides a dynamic and intuitive way to view reports on all available data, or apply Search Criteria to limit the reports for specific purposes and environmental assessment.

Figure 2   CM Reporting Server Web interface supports auditing
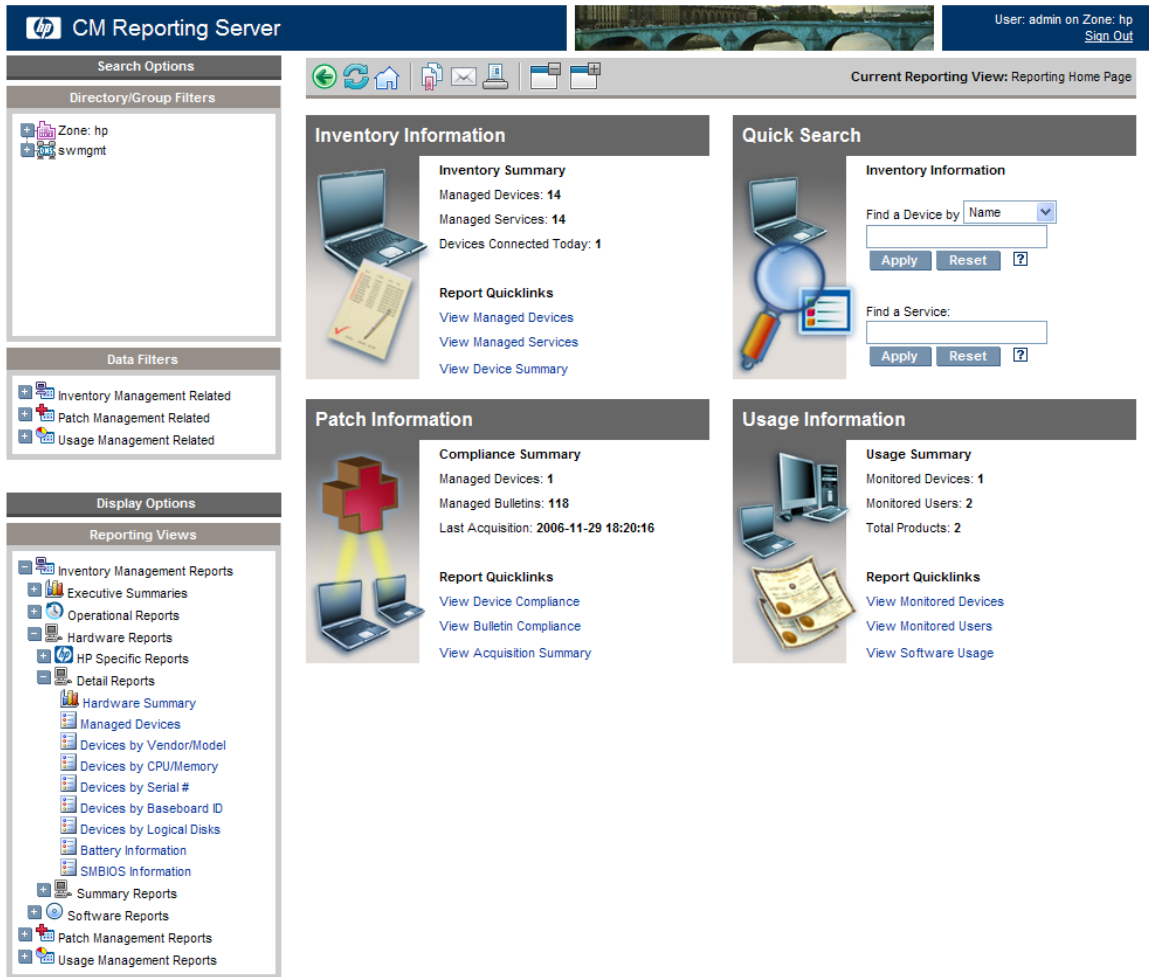
Refer to the *CM Reporting Server Guide* for more information on how to install the CM Reporting Server and create a reporting environment for your CM-related SQL databases, such as inventory, patch, and usage, which can link also to an optional LDAP directory.

The *CM Reporting Server Guide* also explains how to use the CM Reporting Server interface to create, filter, and export reports, as well as browse reports for specific entries.

CM Inventory Manager Terminology

agent computer

(noun) the computer on the end user's desktop that has the CM agent software installed on it.

CIM (Common Information Model)

(noun) 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.

clean machine

(noun) a desktop computer on which the operating system has just been installed, and no further changes have been made.

CM agent

(noun, capitalized) the CM software component that is installed on the end user's desktop computer.

CM Portal

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

CM Messaging Server

The CM Messaging Server is the CM 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 Inventory, Patch, and Portal data collected from CM agents to the appropriate external location. Data Delivery Agents are used to post data directly to an SQL-compliant database using ODBC.

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.

subscriber

(noun) the person (end user) who uses CM-managed applications on a remote desktop computer (CM agent computer).

Web-Based Enterprise Management (WEBM)

Web-Based Enterprise Management 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.

Windows Management Instrumentation (WMI)

Windows Management Instrumentation (WMI) is Microsoft's implementation of WBEM for Microsoft Windows platforms.

WMI Repository

WMI repository is a central storage area designed to hold managed information.

CM Prerequisites

The CM Inventory Manager requires the following CM components:

- CM Configuration Server
- CM agent
    — CM Application Manager

    and/or

    — CM Application Self-service Manager
- CM Messaging Server. See 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 a CM Inventory Manager Database.

Necessary Skills

With CM Products

This document assumes that the reader is familiar with the CM-CSDB, with administering it using the CM Admin CSDB Editor and the CM Portal. Refer to the *CM Administrator Guide* and the *CM Portal Guide* for more information.

With Web-Based Enterprise Management

This document assumes that the reader is familiar with Web-Based Enterprise Management (WBEM). Resources for familiarizing yourself with WBEM can be found at the following web site: **http://www.dmtf.org/spec/wbem.html**.

With Microsoft Implementations of WBEM

This document also assumes that the reader is familiar with Windows Management Instrumentation (WMI). Information concerning WMI can be found at the following web site: **http://msdn.microsoft.com/library/default.asp?url=/library/en-us/w98ddk/hh/w98ddk/wmi_wp_03se.asp**.

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 that is provided below is intended to give you a preliminary understanding of the technology behind the CM Inventory Manager agent. As indicated in Necessary Skills above, we recommend you become more familiar with web-based technology.

Common Information Model (CIM)

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

- Defining **objects**, such as computer systems, devices, controllers, software, files, people, and so forth.

- 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).

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).

Windows Management Instrumentation (WMI)

Windows Management Instrumentation (WMI) is the Microsoft implementation of the Web-Based Enterprise Management (WBEM) that supports the CIM model as well as Microsoft-specific extensions of CIM. To put it simply, it is a set of services designed to input data into a repository via WBEM providers.

The WMI repository is a central storage area designed to hold managed information. It is organized by a series of schemas that are loaded into namespaces. A namespace provides a container, or domain, for the instances of the classes in that schema.

WMI comes installed on Windows 2000 and above. If the module is not installed on a machine, it can be downloaded from the Microsoft Web site at: **http://www.microsoft.com/downloads/details.aspx?FamilyID=afe41f46 -e213-4cbf-9c5b-fbf236e0e875&DisplayLang=en**.

For the purpose of this document, when we refer to WBEM, this includes WMI.

CM and WBEM

The CM Inventory Management agent queries the WBEM namespace (that is, 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 agent. The collected information is then stored in the ODBC inventory database.

Windows 2000 and above systems have WMI natively installed. If necessary, WMI version 1.5 is available for download by licensed end users from the Microsoft download center at **http://www.microsoft.com/downloads/details.aspx?FamilyID=afe41f46-e213-4cbf-9c5b-fbf236e0e875&DisplayLang=en**.

For agent computers with WBEM (Web-Based Enterprise Management) installed, the CM Inventory Manager executes an HP-proprietary method (RIMWBEM) 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 (RIMSFSCAN).

CM Inventory Manager for Window leverages Microsoft's Windows Management Instrumentation (WMI) to collect hardware and software inventory data by using WMI queries. Some WMI queries can traverse the network contacting other servers in the enterprise to collect the requested information. This may result in large volumes of data being returned, and could have a significantly negative effect on network performance. An example of this would be querying all users on the network using the W32_UserAccount WMI class. Extreme caution must be taken to understand the scope of these queries to ensure unexpected results do not occur. While CM Inventory Manager provides an interface to WMI and its providers, it cannot control how these queries are satisfied. It is the customer's responsibility to safeguard against using WMI queries that span the network, if this behavior is not desired.

About this Guide

In addition to this chapter, this book contains the following information.

- **Creating the CM Inventory Manager Environment**
  This chapter describes how to define an ODBC database and DSN, and setup the related CM iInfrastructure servers needed to collect, post, and report the inventory audit information.

- **Installing the CM agents**
  This chapter describes how to install the CM Inventory Agents.

- **The AUDIT Domain**
  This chapter discusses the AUDIT Domain within the CM-CSDB.

- **Software and Hardware Auditing**
  This chapter discusses the different types of auditing, how to configure a supplied audit service, and how to create your own audit service.

- **Configuring Timers for Audit Collection**
  This chapter discusses how to perform audits using a timer.

- **Viewing Inventory from the CM Reporting Server**
  This chapter discusses the topics related to viewing inventory data and reports from the CM Reporting Server.

Summary

- The CM Inventory Manager agent is a utility used to discover hardware and software configuration on remote computers.

- The collection and reporting of inventory information is performed on the CM Inventory Manager when a subscriber/user connects to the CM Configuration Server.

- The CM Messaging Server will automatically create the necessary tables in the SQL or Oracle database you choose.

- The CM Messaging Server uses Data Delivery Agents to deliver the inventory data collected from subscribers to an ODBC-compliant database. Configure the Inventory, Wbem and Core Data Delivery Agents to post data to your ODBC database.

- The CM Reporting Server offers 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, create detailed reports, and filter data using your LDAP directory levels.

- It is suggested that the administrator be familiar with CM products as well as Web-Based Enterprise Management, 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 (that is, the WBEM database) and sends the results back to the CM Configuration Server.

- RIMWBEM queries WBEM namespaces.

- RIMFSCAN directly inspects the software and/or the file system.

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 for the CM Inventory Manager data and an ODBC DSN connection to it.

- Have the following infrastructure installed and configured to support a CM Inventory Manager environment:

  — CM Configuration Server and CM Configuration Server Database (CM-CSDB)

  — CM Administrator CSDB Editor

  — CM Messaging Server

  — CM Reporting Server

  — CM Portal (optional)

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 and its database. See the *HP Configuration Management Configuration Server, Portal and Enterprise Manager 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 to view Inventory Reports. Refer to the *CM Reporting Server Guide*.

❑ Install the CM Admin CSDB Editor. Refer to the *CM Administrator User Guide*.

❑ Optional: Install the CM Portal. Refer to the *CM Portal Guide*.

CM Configuration Server and Database

Before setting up your environment for the CM Inventory Manager, you must have already installed the CM Configuration Server and Database.

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. Microsoft SQL Server 2005 is also supported.

If using Oracle, the minimum database and driver version is Oracle 9i Release 2, patch set 2 (9.2.0.3). Oracle 10g is also supported.

Creating the ODBC Database and DSN for CM Inventory Manager

Before installing the 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 varies based on the number of managed devices in your environment and type of inventory audit information being collected. The procedures below merely reflect recommendations.

To create a Microsoft SQL CM 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 inventorydata.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 Configuration Server media contains a CM Administrator installation. Use it to install the CM Admin CSDB Editor. Refer to the *CM Administrator User Guide* for information on installing and using the CSDB Editor.

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 the following 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

[Explain more here, why there are three DDAs, and tips for improving performance, perhaps adding WORKERS and using store and forward.]

CM Reporting Server

The CM Reporting Server version 5.00 is required to view basic and 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, see 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. Define the database and the DSN connection to it.

- 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 for installing CM agents.

- Know how to customize the installation process.

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

- Be able to modify the [ARGS] section of the INSTALL.INI in order to customize the behavior of the HP Configuration Management Application Self-service Manager (CM Application Self-service Manager).

- Be able to modify the [OBJECTS] section of the INSTALL.INI in order to specify CM objects to be created on the agent computer.

- Know how to use the Installation Wizard.

- Know how to remove, repair, and modify installations of CM agents using the Installation Wizard and command lines.

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

- Know how to use a post-install script to run processes after installing CM agents.

The CM agent installation program uses Microsoft Windows Installer. The program consists of one MSI package with six feature sets:

- HP Configuration Management Application Manager (CM Application Manager)

- HP Configuration Management Application Self-service Manager (CM Application Self-service Manager)

- HP Configuration Management Inventory Manager (CM Inventory Manager)

- HP Configuration Management OS Manager (CM OS Manager)

- HP Configuration Management Patch Manager (CM Patch Manager)

- HP Configuration Management Server Management (CM Server Management)

> ⚠ 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 HP Configuration Management Configuration Server (CM Configuration Server).

System Requirements

- Windows 2000 with SP3, Server 2003 (x86, x64, IA64), XP (x86, x64, IA64) or Vista (x86, x64).

- Microsoft .NET runtime Version 1.1 or higher (required for CM Application Self-service Manager only). The .NET installation program is available in the \DotNet folder on the CM agent media. If .NET does not exist on the agent computer, the .NET installation program runs automatically. Microsoft .NET requires Microsoft Internet Explorer 5.01 or later.

- TCP/IP connection to a computer running the CM Configuration Server.

- You must have administrator rights to the computer to install the CM agents.

Platform Support

For information about the platforms that are supported in this release, see the accompanying release notes.

About the Installation Process

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

- **setup.exe**
  setup.exe is stored in the \win32 directory on the CM agent media. It accepts any standard Windows Installer command line parameters and passes them to the Windows Installer service.

  You can also create an AIP for network installations.

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

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

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

  The target directory contains CM-MGMTAPPS*xx*.MSI (where *xx* represents the first two digits of the latest version number), the installation folders, setup.exe, and any files (such as INSTALL.INI or Visual Basic scripts) stored in the same directory as setup.exe. Next, copy the \DotNet folder into the target directory.

- **CM-MGMTAPPS*xx*.MSI**
  CM-MGMTAPPS*xx*.MSI (where *xx* represents the first two digits of the latest version number) is the MSI database file, which contains the default configuration information for the installation. This file is stored in the \win32 directory on the CM agent media.

- **INSTALL.INI**
  Use INSTALL.INI to customize the installation or the CM agent arguments file, or to create or set attributes for CM objects. Settings in INSTALL.INI override the defaults stored in CM-MGMTAPPSxx.MSI.

A sample `INSTALL.INI` is available in the `\win32` directory on the CM agent media.

- **args.xml**
  `args.xml` is the CM Application Self-service Manager arguments file created from information stored in the [ARGS] section of `INSTALL.INI`. This file, stored in `IDMLIB` on the agent computer, controls the behavior of the CM Application Self-service Manager. The default directory for `IDMLIB` is
  SystemDrive`:\Program Files\Hewlett-Packard\CM\Agent\Lib\`.

- **Pre-install scripts**
  (*Recommended for experienced users only*.)
  Use custom Visual Basic scripts to customize MSI properties that affect the installation. For an example of a simple script, see Using a Pre-Install Script on page 66.

- **Post-install scripts**
  (*Recommended for experienced users only*.)
  Use custom Visual Basic, REXX, or Tcl scripts to run processes such as the first agent connect. For an example, see Using a Post-Install Script on page 67.

Figure 3   CM agent installation process
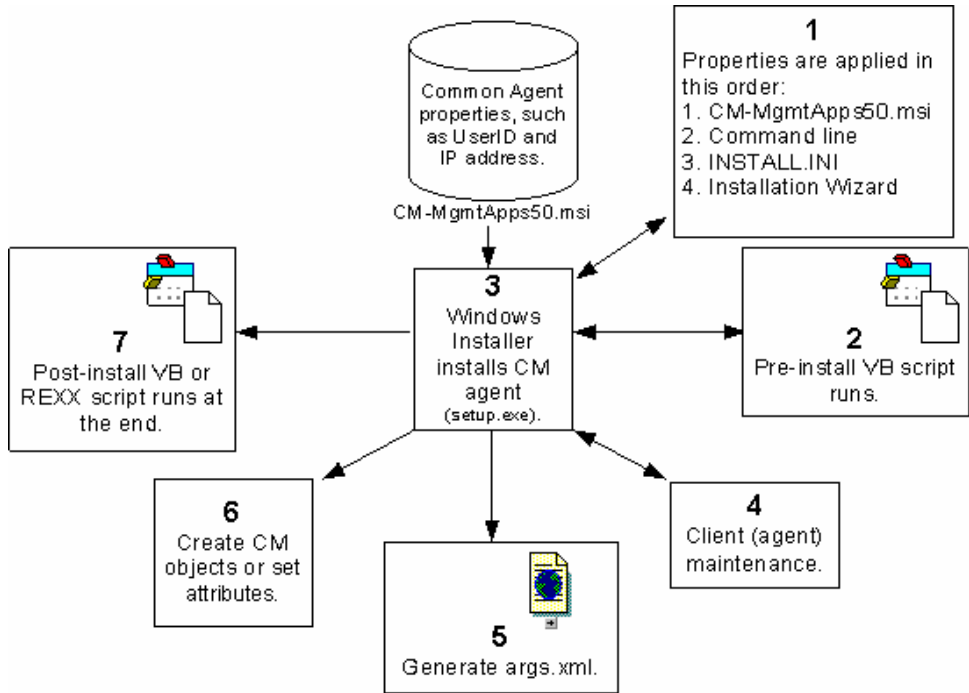
Preparing INSTALL.INI

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

- Customize the installation.
- Customize the CM Application Self-service Manager.
- Create or set attributes for CM objects.

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

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

The following is a sample INSTALL.INI

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

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

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

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

;Uncomment to show/hide panels in the installation wizard (defaults shown below)
;NVDENABLEUSER=Y
;NVDENABLEIP=Y
;NVDENABLEPROXY=N
;NVDENABLESHORTCUT=Y
;NVDSHORTCUT=Y
;NVDSTARTMENUICON=Y
;NVDSTARTWMICFGMGR=Y
;NVDRAMSHORTCUT=N
;NVDRAMSTARTMENUSHORTCUT=N
;NVDRAMCONNECT="radntfyc localhost radskman
        cat=prompt,uid=$machine,ulogon=n,hreboot=y,ind=y,ask=y,ip=xxx.xxx.xxx.xxx,port=3464,mnam
        e=Radia,dname=Software,startdir=SYSTEM,rtimeout=1800,context=m,log=connect_manual.log"

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

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

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

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

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

[Args]
; Tags to be placed into the CM Application Self Service Manager Agent ARGS.XML file
; A value of _NONE_ removes the tag from the file
```

```
;
;askconfirm=
;channelname=software
;dataurl=
;enterprisemanagement=
;identification=$USER
;log=connect.log
;logsize=
;logonpanel=
;managerurl=
;providername=Radia
;redirect=
;resolutionmanager=
;resolutionport=
;sslmanager=
;sslport=
;startdir=
;uioption=

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

[PROPERTIES] Section of INSTALL.INI

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

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

Table 4   [PROPERTIES] section of INSTALL.INI

Argument	Description	Default Value*
ADDLOCAL	Specify the features that you want to install on the local hard drive. You must be properly licensed to use these products. The features may be: • NVDINSTALLRAM = Application Manager • NVDINSTALLRSM = Application Self-service Manager • NVDINSTALLRIM = Inventory Manager • NVDINSTALLROM = OS Manager • NVDINSTALLPATCH = Patch Manager • NVDINSTALLSVR = Server Management • NVDINSTALLRLAE = CM Local AIP Extension Note: This must be installed to use the Local AIP support for the CM MSI Redirector feature. Refer to the *CM Application Manager and Application Self-Service Manager Guide* for more information.	N/A

Argument	Description	Default Value*
ARPNOREMOVE	Set ARPNOREMOVE to 1 to indicate that you want to disable the ability to remove the CM agent from the computer using Add/Remove Programs in the Control Panel. Note: Setting ARPNOREMOVE to 0 will *not* disable this option due to a Windows Installer issue. If you want to allow your subscribers to remove the CM agent from the computer using Add/Remove Programs in the Control Panel, place a semi-colon (;) in front of the ARPNOREMOVE argument in `INSTALL.INI`. • For Windows 2000 or XP, the Remove button is disabled. • For earlier operating systems, the CM agent will not be listed in Add/Remove Programs in the Control Panel.	Subscribers can remove the CM agent from the computer using Add/Remove Programs in the Control Panel.
INSTALLDIR	Specify the directory where you want to install the CM agent. This value will be overridden if a new directory is specified in the Destination Folder window in the CM Agent Installation Wizard.	`SystemDrive:\Program Files\Hewlett-Packard\CM\Agent`

Argument	Description	Default Value*
NVDENABLEUSER	Indicate whether to show or hide the Set User window in the Installation Wizard. • Y – Show the window. • N – Hide the window. Note: If you hide the window, the Create CM Application Self-Service Manager icon on the desktop check box will no longer be available to your subscribers. • D – Show the window, but disable the User Name field. The Create CM Application Self-Service Manager icon on the desktop check box is still available.	Y
NVDENABLEIP	Indicate whether to show or hide the CM Configuration Server window in the Installation Wizard.	Y
NVDENABLEPROXY	Indicate whether to show or hide the Proxy Information window in the Installation Wizard. If you want to use a CM Proxy Server during the CM agent connect, show this window. The information entered in the Proxy Information window is stored, by default, in the PROXYINF object in the agent computer's IDMLIB directory.	N

Argument	Description	Default Value*
NVDENABLESHORTCUT	Indicate whether to show the Create CM Application Self-Service Manager icon on the desktop check box in the Set User window. Selecting this check box installs a desktop shortcut for the CM Application Self-service Manager on the subscriber's desktop.	Y
NVDSHORTCUT	Indicate whether to install a desktop shortcut for the CM Application Self-service Manager on the subscriber's computer.	Y
NVDSTARTMENUICON	Indicate whether to install an icon in the Start Menu for the CM Application Self-service Manager on the subscriber's computer.	Y
NVDSTARTWMICFGMGR	Indicates whether to install the shortcuts for WMI.	Y
NVDRAMSHORTCUT	Indicate whether to install a desktop shortcut for the CM Application Manager on the subscriber's computer.	N
NVDRAMSTARTMENUSHORTCUT	Indicate whether to install an icon in the Start Menu for the CM Application Manager on the subscriber's computer.	N
NVDRAMCONNECT	Specify a command line to run if a CM Application Manager shortcut is created on the desktop or the Start Menu.	Blank

Argument	Description	Default Value*
NVDMAINTDIR	Specify a directory that stores the CM agent maintenance files. Note: Type a value only if you want to store maintenance files in a directory other than the `MAINT` subdirectory in the folder containing setup.exe. If files exist in this directory that are newer than the installation files, they will be copied into the CM agent's `IDMSYS` directory.	`MAINT` subdirectory in the folder containing `setup.exe`
NVDLOCALNOTIFYONLY	If set to Y, the agent will allow CM Notifies only from the local host.	N
NVDRADTRAYSTART	Set to Y to start the System Tray automatically, if the CM Application Manager is selected during the agent installation process.	N
NVDNOTIFYINTERACT	Set to Y to enable the HP OVCM Notify Daemon to interact with the desktop.	N
NVDREDIRECTORINTERACT	Set to Y to enable the HP OVCM MSI Redirector to interact with the desktop.	N
NVDSCHEDULERINTERACT	Set to Y to enable the HP OVCM Scheduler to interact with the desktop.	N

Argument	Description	Default Value*
NVDPRECAPATH	Specify the fully qualified path and filename of a custom Visual Basic pre-install script. Note: New objects or properties must be defined in INSTALL.INI. You can use a pre-install script to override a value for the object or property, but if you attempt to specify a new object or property in the pre-install script, it will be ignored. For an example of a simple script, see Using a Pre-Install Script on page 66.	N/A
NVDPOSTCAPATH	Specify the fully qualified path and filename of a custom Visual Basic or REXX post-install script. For an example, see Using a Post-Install Script on page 67.	N/A

[ARGS] Section of INSTALL.INI

Use the [ARGS] section to control the behavior of the CM Application Self-service Manager. The information in this section is used to build the CM Application Self-service Manager arguments file, args.xml, which is stored in IDMLIB on the agent computer. The default directory for IDMLIB is *SystemDrive*:\Program Files\Hewlett-Packard\CM\Agent\Lib\.

Below is an example of args.xml.

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

```
<RESOLUTIONPORT>3464</RESOLUTIONPORT>

</ARGUMENTS>

</RADIA_ARGUMENTS>
```

> The XML tags (arguments) described in this section are not case-sensitive when you type them in INSTALL.INI. However, the arguments will automatically be converted to all uppercase in args.xml.
>
> If you are using the CM Application Manager, any of the parameters in the [ARGS] section can be added to the radskman command line.

Table 5   [ARGS] Section of INSTALL.INI

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

Argument	Mandatory or Optional	Description	Default Value
enterprisemanagement	Optional	*For use with the CM Application Self-service Manager only.* Deploys mandatory applications from the CM Application Self-service Manager. Set Enterprisemanagement =auto to deploy mandatory applications. Remove the Enterprisemanagement tag if you do not want to deploy mandatory applications.	N/A
identification	Optional	Identifies the agent to the CM Configuration Server by defining the value for the ZUSERID variable in the ZMASTER object. This value will be overridden if a different User Name is specified in the Set User window in the CM Agent Installation Wizard. If you do not want this value to be modified, set NVDENABLEUSER = N in the [PROPERTIES] section of INSTALL.INI. **$MACHINE** – The CM user ID is the name of the subscriber's computer. **$USER** – The CM user ID is the logon ID for the subscriber currently logged on.	$USER

Argument	Mandatory or Optional	Description	Default Value
		CUSTOM – literal custom specification.	
log	Optional	Specifies the name of the log stored in IDMLOG. IDMLOG is specified in `NVD.INI`. `NVD.INI` is stored in *SystemDrive*:`\Program Files\Hewlett-Packard\ CM\Agent\LIB,` by default.	`Connect.log`
logsize	Optional	Specifies the size of the log file in bytes. When the logsize is reached, a backup file (`.BAK`) is created. By default, this file is `connect.bak`. If a backup file already exists, it will be overwritten.	1000000
logonpanel	Optional	Controls the display of the logon panel.	Y
managerurl	Optional	Specifies the address of the CM Configuration Server to be used for HTTP object transfer. The managerurl specification takes the form `http://`*hostname*`:`*port*`/nv durl`.	N/A
providername	Mandatory	The name of the CM Configuration Server, which is set during the standard installation. This is used to name the folder below the STARTDIR on the agent computer. See startdir on page 51 for more information.	Radia

Argument	Mandatory or Optional	Description	Default Value
redirect	Optional	*Used for the CM Application Self-service Manager only.* Specifies an alternate start-up file (`filename.xml`), which can be accessed via a network path or URL. If the redirect tag is set in `args.xml`, the CM Application Self-service Manager uses the properties specified in the alternate file. For example, you might set redirect to point to a start-up file on the network that is shared by all users.	N/A
resolutionmanager	Mandatory	Identifies the IP address for the CM Configuration Server. You can also use the server name. This value will be overridden if a different IP address is specified in the CM Configuration Server window in the CM Agent Installation Wizard. If you do not want this value to be modified, set NVDENABLEIP = N in the [PROPERTIES] section of the INSTALL.INI.	N/A
resolutionport	Mandatory	Identifies the port for the CM Configuration Server. This value will be overridden if a different port is specified in the CM Configuration Server window in the CM Agent	N/A

Argument	Mandatory or Optional	Description	Default Value
		Installation Wizard. If you do not want this value to be modified, set NVDENABLEIP = N in the [PROPERTIES] section of the INSTALL.INI.	
sslmanager	Optional	Specifies the address of the CM Configuration Server to be used for SSL communications. If you want self-maintenance to use SSL communications, concatenate ::SM to the end of the specified IP address or host name. For example, sslmanager=*hostname*::SM Warning: Use the ::SM switch with the following caveat in mind; the file (cacert.pem) that contains the CA root certificates cannot be maintained. If the corresponding CA root certificate for the certificate in use by the CM Configuration Server should ever become expired, revoked, or corrupt, it will result in disabling SSL communications to the CM Configuration Server.	N/A
sslport	Optional	Specifies the TCP/IP port that the SSL manager is listening on. This is normally 443. The sslport specification takes the form *sslport=port*.	N/A

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

[OBJECTS] Section of INSTALL.INI

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

Table 6   [OBJECTS] section of install.ini

Argument	Description	Default Value
ZMASTER_ZDSTSOCK	The port setting for the CM Configuration Server.	3464

Argument	Description	Default Value
ZMASTER_ZIPADDR	The IP address for the CM Configuration Server.	N/A
ZMASTER_ZNTFPORT	The port that the CM agent's notify daemon is 'listening' on.	3465
ZMASTER_ZNTFYSEC	This attribute allows a Notify operation to execute programs only from the IDMSYS directory. This is used for security during Notify operations.	Y
ZMASTER_ZTIMEO	Amount of time that the CM agent will wait for a response from the CM Configuration Server before the CM agent program is deactivated (timed out). Numeric value between 0 and 3200, in seconds.	240
ZMASTER_ZTRACE	Enables log to include communications buffer information; also generates unique logs for create methods. **Y** Turns on Communication and Client Method Tracing. **S** Turns on Communication summary information; Client Method Tracing is *not* enabled. **N** Communication Tracing and Client Method Tracing are disabled.	N
ZMASTER_ZTRACEL	Level of tracing generated in the CM agent log files. 0 to 999, where 0=minimal tracing, 40=acceptable for most activity, and 999=maximum.	040
ZMASTER_ZUSERID	The subscriber's user ID.	The name of the user currently logged on to the computer.

Argument	Description	Default Value
ZMASTER_ZVRFYUID	Verify the user ID sent by the CM Configuration Server's Notify command with the ZUSERID field found in the CM agent's ZMASTER object.	N
PROXYINF_USEPROXY	Indicates whether you use a proxy server to connect to the CM Configuration Server.	N
PROXYINF_DISCOVER	*For use with Microsoft Internet Explorer.* Set to the proxy address and port that your web browser is using.	N
PROXYINF_PROXADDR	The IP address and port number of your proxy server.	xxx.xxx.xxx.xxx:1080
RADSETUP_COP	Set this to Y to enable Client Operations Profiles.	N

Installing the CM Agents

Before installing the CM agents, you must decide whether to initiate the installation from:

- a command line that initiates the installation from a web page, FTP site, mapped drive, CD-ROM, or e-mail. See Installing the CM Agent from a Command Line on page 54.

- a logon script. See Initiating the CM Agent Installation from a Logon Script on page 56.

- the CM Portal. This is recommended for mass rollouts in a Windows environment. Refer to the *CM Portal Guide* for more information.

After initiating the installation, the CM Agent Installation Wizard runs. This section describes some of the ways that you can initiate the CM agent installation, and then describes the standard CM Agent Installation Wizard.

Installing the CM Agent from a Command Line

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

- how you are going to make the CM agent installation program available to your subscribers. You can do this via a Web page, FTP site, mapped drive, CD-ROM, or e-mail.

- what CM agent features (CM Application Manager, CM Application Self-service Manager, CM Patch Manager, CM Inventory Manager, CM OS Manager, CM Server Management) you want to install and pass any additional arguments to the command line.

Example

The following is an example of a command line that will install the CM Application Self-service Manager silently and create a detailed Windows Installer log.

```
SETUP.EXE ADDLOCAL=NVDINSTALLRSM /qn /L*v C:\Program
Files\Hewlett-Packard\CM\Agent\install.log
```

The arguments in this command line, and others, are described in Table 7 below and Table 8 on page 55.

Specifying the CM Agent Features to Install

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

Table 7   CM agent feature state arguments

Specify the following arguments	To set the feature state
ADDLOCAL	Type a comma-delimited list of features that you want set to "Will be installed on local hard drive."

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

When specifying features on the command line, reference the CM agent features as follows:

- NVDINSTALLRAM installs the CM Application Manager.

- NVDINSTALLRSM installs the CM Application Self-service Manager.

- NVDINSTALLRIM installs the CM Inventory Manager.

- NVDINSTALLROM installs the CM OS Manager.

- NVDINSTALLPATCH installs the CM Patch Manager.

- NVDINSTALLSVR installs CM Server Management.

For example, to install the CM Application Manager and the CM Application Self-service Manager to the computer, use the following command line:

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

Additional Command Line Arguments

Some additional arguments that you can pass to the installation program on the command line are described Table 8 below.

Table 8   Command Line Arguments

Sample	Sample
/qn	Performs a silent installation.
	Note: A silent installation is one that takes place without a user interface. Throughout our literature, the terms "silent installation," "quiet installation," and "unattended installation" are all used to refer to the same process.

Sample	Sample
`/qb`	Displays the progress bar only during the installation.
`/L*v` `drive`:`\install.log`	Creates a detailed Windows Installer log. Note: Using this option may impact the performance of the installation.
`/a TARGETDIR=` `drive`:`\targetdirectory`	Creates a Windows Installer AIP in the specified target directory. Note: A Windows Installer Administrative Installation Point (AIP) is also known as an Administrative Control Point (ACP). The target directory contains `CM-MGMTAPPSxx.MSI`, the installation folders, `setup.exe`, and any files (such as `INSTALL.INI` or Visual Basic scripts) stored in the same directory as setup.exe. Once you have created the AIP, you can run `setup.exe` and pass the appropriate command line parameters. This starts the Windows Installer and passes the specified parameters to it.
`NVDINIFILE=path` `\INIfilename`	To rename the installation INI file, pass this parameter to the command line. Be sure to include the fully qualified path. By default, the installation program refers to `INSTALL.INI` located in the current directory.
`INSTALLDIR=`	Specify the installation directory. Use quotes if the path contains spaces.

If you initiate a CM agent installation with a command line that does not contain the silent installation arguments (`/qn`), the CM Agent Installation Wizard opens. See for more information.

Initiating the CM Agent Installation from a Logon Script

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

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

The following is an example of code that you can add to the logon script that installs the CM agents. If the CM agents are *not* already installed when the subscriber logs on to the server, this logon script runs the CM agent installation program.

Here is a sample logon script:

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

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

> The ZMASTER object begins the resolution process and is the first object exchanged during the agent connect process.

If ZMASTER.EDM *exists*, the script skips the installation. If ZMASTER.EDM *does not* exist, the CM agent installation program launches.

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

This command line performs a silent installation of the CM agents.

> Modify this script to reflect your organization's needs.

If you initiate a CM agent installation using a command line that does not contain the silent installation arguments, the CM Agent Installation Wizard opens. See Using the CM Agent Installation Wizard on page 58 for more information.

Using the CM Agent Installation Wizard

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

To install CM agents using the Installation Wizard

2   From the folder containing the CM agent installation files, run setup.exe.

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

The CM Agent Installation Wizard opens.

2   Click **Next**.

The License Agreement window opens.

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

The Destination Folder window opens. The default location for the CM agents is *SystemDrive*:\Program Files\Hewlett-Packard\ CM\Agent.

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

4   Click **OK** to continue.

5   Click **Next**.

The Set User window opens.

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

7   Select the **Create CM Application Self-Service Manager icon on the desktop** check box if necessary.

8   Click **Next**.

The CM Configuration Server window opens.

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

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

11 Click **Next**.

    The Select Features window opens.
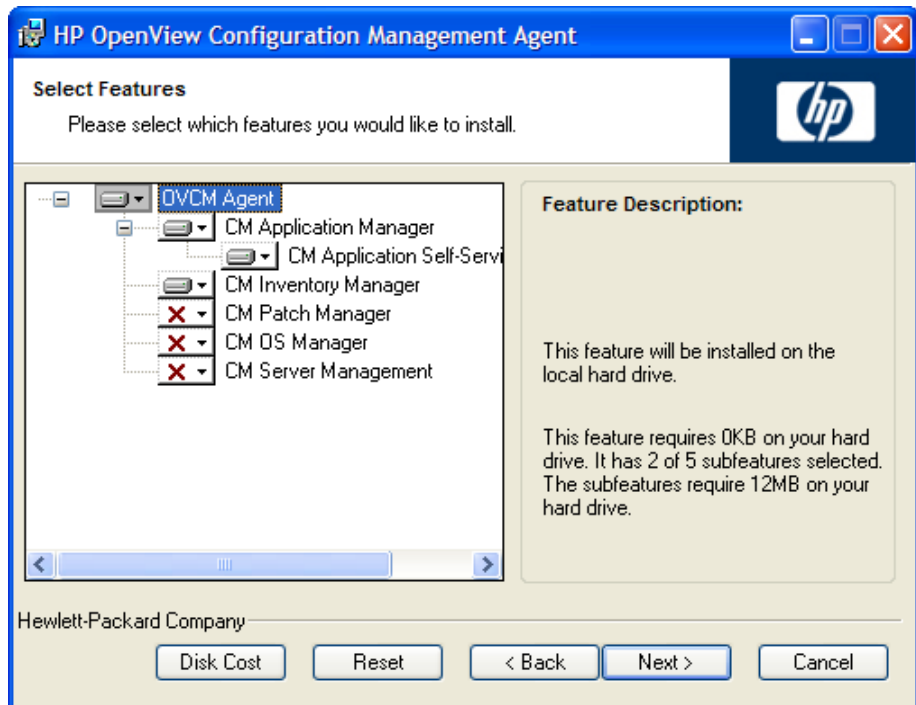
12 Click ▣▾ to select the features that you want to install.

    Each time you click ▣▾, a shortcut menu for that feature opens.

    ▶   Install only the CM agents for which you are licensed.

13 From the shortcut menu, select an installation option. These options are
    described in Table 9 on page 60.

Table 9   **Feature settings for CM agents**

Option	Description
Will be installed on local hard drive	Installs the top-level feature on the local hard drive, but not any sub-features listed below.
Entire feature will be installed on local hard drive	Installs the entire feature, including any sub-features listed below. Note: In this installation program, selecting this option or the "Will be installed on local hard drive" option for the CM Application Self-service Manager, CM Application Manager, or CM Inventory Manager results in the same installation because these features do not contain sub-features. Example: If you select this option for the CM agent feature in the Select Features window, all of the CM agents will be installed on the local hard drive.
Entire feature will be unavailable	The feature will not be installed. If previously installed this feature will be removed.

> If you want to set the same options for all of the features, you can click ▣▾ **OVCM Agent** and select the appropriate option to apply the setting to all features.
>
> Click **Disk Cost** to see an overview of the disk space needed for the installation.

14 Click **Next**.

If .NET is not installed on the agent computer and you have chosen to install the CM Application Self-service Manager, .NET will be installed during the CM agent installation. However, if you copied the installation program to your computer and did not include the \DotNet folder, the DotNet Settings message in the next figure opens.

15 Click **OK**.

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

If .NET is not already installed on the computer, the .NET Installation window opens.

17 Click **Next**.

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

The Ready to Install the Application window opens.

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

If necessary, the .NET Framework Setup wizard opens. Follow the prompts to install .NET on the agent computer. After .NET is successfully installed, the CM agent installation begins.

When the installation is done, successful installation window opens.

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

Removing the CM Agents

The Windows Installer installation program offers the ability to remove your CM agents. This section describes how to remove the CM agent using the Installation Wizard and using a command line.

Using the Installation Wizard to Remove CM Agents

This section describes how to remove the CM agent using the Installation Wizard.

⚠️ To remove specific features of the CM agent, use the Modify option on the Application Maintenance window. This is discussed in Modifying the CM Agent Installation on page 64.

To remove CM agents using the Installation Wizard

1 From the folder containing the CM agent installation files, double-click **setup.exe**. The Application Maintenance window opens.

2 Select the **Remove** option.

3 Click **Next**.

The CM Agent Uninstall window opens.

4 Click **Remove.**

The files for all CM agents are removed from the computer.

The CM agent has been successfully uninstalled window opens.

5 Click **Finish**.

Using a Command Line to Remove CM Agents

This section describes how to remove CM agents using a command line.

- From the folder containing the CM agent installation files, type the following command line:

      SETUP.EXE REMOVE=ALL

  See Installing the CM Agent from a Command Line on page 54 for additional arguments.

  or

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

Reference the features for the CM agent installation as follows:

- CM Application Manager   NVDINSTALLRAM
- CM Application Self-service Manager   NVDINSTALLRSM
- CM Inventory Manager   NVDINSTALLRIM
- CM OS Manager   NVDINSTALLROM
- CM Patch Manager   NVDINSTALLPATCH
- CM Solutions for Servers   NVDINSTALLSVR

Example

To silently remove the CM Application Self-service Manager and CM Application Manager, type:

SETUP.EXE REMOVE=NVDINSTALLRSM,NVDINSTALLRAM /qn

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

Repairing the CM Agents

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

This section describes how to repair CM agents using the Installation Wizard and using a command line.

Using the Installation Wizard to Repair CM Agents

This section describes how to repair CM agents using the Installation Wizard.

To repair CM agents using the Installation Wizard

1   From the folder containing the CM agent installation files, double-click **setup.exe**.

    The Application Maintenance window opens.

2   Select the **Repair** option.

3   Click **Next**.

    The Ready to Repair the Application window opens.

4   Click **Next**.

    When the repair is done, the CM agent has been successfully installed window opens.

5   Click **Finish**.

Using a Command Line to Repair CM Agents

This section describes how to repair CM agents using a command line.

To repair CM agents using a command line

•   From the folder containing the CM agent installation files, type the following command line:

```
msiexec /f CM-MGMTAPPSXX.MSI
```

Where *xx* represents the first two digits of the latest version number.

> ➤ You can use additional parameters with this command line. For more information, see your Windows Installer documentation.

Modifying the CM Agent Installation

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

Using the Installation Wizard to Modify the Installation of CM Agents

This section describes how to modify the installation of CM agents using the Installation Wizard.

To modify the installation of CM agents using the Installation Wizard

1 From the folder containing the CM agent installation files, double-click **setup.exe**.

   The Application Maintenance window opens.

2 Select the **Modify** option.

3 Click **Next**.

   The Select Features window opens. See Using the CM Agent Installation Wizard on page 58 for information about how to use this window.

4 Click **Next**.

   The Ready to Modify the Application window opens.

5 Click **Next**.

   The CM agent has been successfully installed window opens.

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

Using a Command Line to Modify the Installation of CM Agents

This section describes how to modify the installation of CM agents using a command line.

To modify the installation of CM agents using a command line

- From the folder containing the CM agent installation files, type the following command line:

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

Table 10   CM agent feature state arguments

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

Reference the features for the CM agent installation as follows:

- CM Application Manager   NVDINSTALLRAM
- CM Application Self-service Manager   NVDINSTALLRSM
- CM Inventory Manager   NVDINSTALLRIM
- CM OS Manager   NVDINSTALLROM
- CM Patch Manager   NVDINSTALLPATCH
- CM Solutions for Servers   NVDINSTALLSVR

If you want to install the CM Application Self-service Manager, and to make the CM Inventory Manager and CM Application Manager unavailable, use the following command line:

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

See Installing the CM Agent from a Command Line on page 54 for additional arguments.

Using a Pre-Install Script

Use Visual Basic scripts to customize MSI properties that affect the installation. The following is a very simple Visual Basic script, which is intended to be an example only.

⚠️ Be sure to use the NVDPRECAPATH argument to specify the fully qualified path and file name of a custom Visual Basic pre-install script in INSTALL.INI or on the command line. See the description of NVDPRECAPATH in Table 4 on page 40.

Here is a sample pre-install script:

```
' The following sample demonstrates fetching an MSI property,
then setting the same property.

' The property values are displayed in message boxes for
debugging purposes.

Option Explicit

msgbox Session.Property("ALLUSERS")

Session.Property("ALLUSERS") = "1"

msgbox Session.Property("ALLUSERS")
```

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

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

To override property settings or attributes for objects

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

- Use the prefix NVDARG to override property settings.

  For example, to override the value set for the identification property, which identifies the subscriber session to the CM Configuration Server, type:

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

- Use the prefix NVDOBJ to override object attributes.

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

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

Using a Post-Install Script

Use custom Visual Basic, REXX, or Tcl scripts to run processes after installing CM agents. For example, your post-install script can initiate a connection to the CM Configuration Server in order to process mandatory applications.

⚠ Be sure to use the NVDPOSTCAPATH argument to specify the fully qualified path and filename of the custom Visual Basic or REXX post-install script in INSTALL.INI or on the command line. See the description of NVDPOSTCAPATH in Table 4 on page 40. For example, if you want to run a script called redstart.rex, uncomment and set NVDPOSTCAPATH=C:\Progra~1\Hewlett-Packard\radstart.rex

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

The following is a sample REXX from a post install script.

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

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

Summary

- The CM agent installation program consists of one package with six feature sets: CM Application Manager, CM Inventory Manager, and CM Application Self-service Manager, CM Patch Manager, CM OS Manager, CM Server Management.

- You can install the CM agents using a command line or using the Installation Wizard.

- Customize your installation to suit your enterprise.

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

4 The AUDIT Domain

At the end of this chapter, you will:

- Understand the AUDIT Domain in the CM Configuration Server Database (CM-CSDB).

- Understand the CM Inventory Manager database.

This manual is provided to assist you with installing and using the CM Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

The AUDIT Domain in the CM Configuration Server Database

The AUDIT Domain is located in the PRIMARY File of the CM-CSDB. The AUDIT Domain contains the classes required to:

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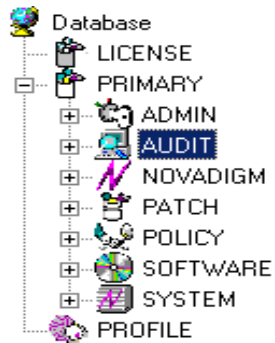
AUDIT Domain Defined

The AUDIT Domain is structured very much like the SOFTWARE Domain. The next figure shows its tree structure in the CM Admin CSDB Editor.
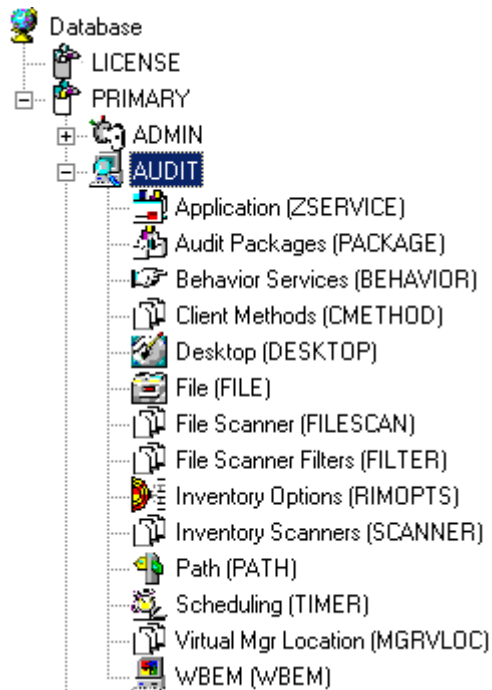
Table 11 below describes the classes in the AUDIT Domain.

Table 11   AUDIT Domain

Class	Description
Application (ZSERVICE)	These are 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.

Class	Description
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)	This class is 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 DLLs.
File Scanner (FILESCAN)	For UNIX devices only, persistent component class used to configure an inventory scan. Adding File Scanner components to an audit package creates instances of the FILESCAN Class.
File Scanner Filters (FILTER)	For UNIX devices only, 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 that control an inventory management task. For additional information, refer to the RIMOPTS Class on page 75.
Inventory Scanners (SCANNER)	This persistent component class is used to configure an inventory scan. Create instances of the SCANNER Class by adding Inventory Scanners components to an audit package.
Path (PATH)	This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Registry (REGISTRY)	This class 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.

Class	Description
Scheduling (TIMER)	This class 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.
Virtual Mgr Location (MGRVLOC)	This class is used to specify the initial path for files being transferred to the CM Configuration Server during a FILE audit.
WBEM (WBEM)	This class contains instances that define CM 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.

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 12 below describes these attributes.

Table 12   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.

Attribute	Usage
RUNEXEC	This string 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 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 additional information about editing instances, refer to the CM Admin CSDB Editor topics in the *CM Administrator User 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 (8) (ZLIST)** → **CONNECT_** → **CONNECT_ZSERVICE_**

2   Double-click on **CONNECT_ZSERVICE_TO_RULES**.
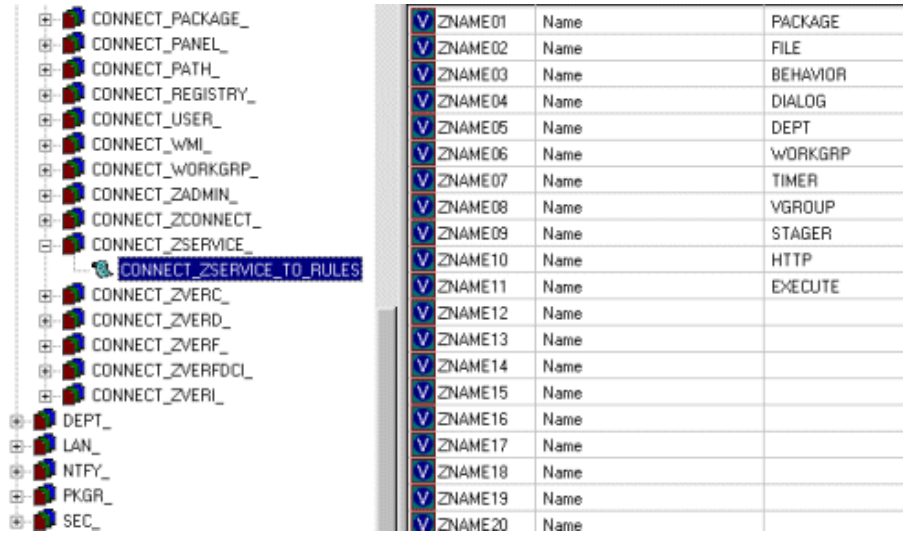
3　The Editing Instance dialog box opens.
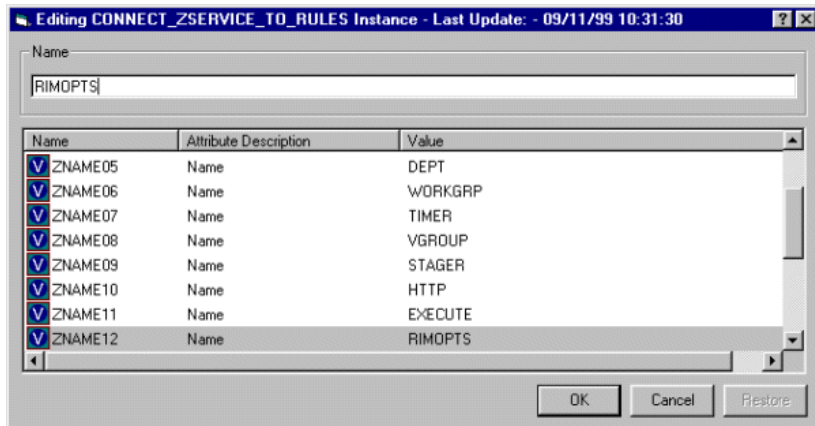
4　Set the value of the **ZNAME n** attribute to **RIMOPTS**.

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

REGISTRY Class

The Registry Class uses WMI to obtain a Registry scan of a Windows machine. Most of the attributes are copied from the existing WBEM class of the AUDIT Domain, with descriptions adjusted for registry-specific needs. For example, the PROPERTY and CNDITION attributes define the current

Registry hive and subkey to scan, respectively. Three new Registry-specific attributes have been added to the class. They include:

- **RPTCLASS** – The Report Class Name in RIM.

- **FORMAT** – The Output format- requires REGISTRY (do not change).

- **DEPTH** – Defines the levels below the current subkey to scan.

Table 13 below summarizes the attributes and values for the Registry Class instances. Attributes in bold are new to this class (not in the WBEM class).

Table 13   Registry Class Instance Attributes

Attribute	Description	Default Value	Valid Values
ACTION	Report Flags (I, N, C, D, S, D, C)	YYYYXXN	Y, X, or N for each flag.
NAMSPACE	Name Space	root\default	root\default – Do not change.
CLASS	WBEM Class	StdRegProv	StdRegProv – Do not change.
RPTCLASS	Report Class Name	Registry	A valid table name. If blank "StdRegProv" will be used.

PROPERTY	Registry hive.	HKEY_LOCAL_MACHINE	Any Windows registry hive:
			`HKEY_CLASSES_ROOT`
			`HKEY_CURRENT_USER`
			`HKEY_LOCAL_MACHINE`
			`HKEY_USERS`
			`HKEY_CURRENT_CONFIG`
			`HKEY_DYN_DATA`

CNDITION	Registry subkey	SOFTWARE\Microsoft\Internet Explorer	Any Windows registry subkey.
FORMAT	Output format.	REGISTRY	REGISTRY – Do not change.

Attribute	Description	Default Value	Valid Values
DEPTH	Starting at the registry subkey named in the CNDITION attribute, depth specifies the number of descendent key levels to include in the scan.	0	0, -1, or n • Set to **0** to only scan current subkey. • Set to **-1** to scan all subkey levels. • Set from **1** through n to scan the current subkey and the specified number of subkey levels deep.
OUTPUT	Output Object Name	WBEMAUDT	WBEMAUDT
TYPE	Scan Type (WBEM)	WBEM	WBEM – Do not change.
NAME	Friendly Name	Default	Friendly name for this instance displayed in the CM Admin CSDB Editor.

Implementing Registry Scans

Use the following high-level procedures to create and run scans of the Windows Registry using the REGISTRY class in the AUDIT Domain.

1   Create an AUDIT.PACKAGE instance for the registry scan.

2   Right-click on the newly created AUDIT.PACKAGE instance and select **Add Component** from the shortcut-menu.

3   Use the Add Component dialog to both create and edit a new AUDIT.REGISTRY instance is a few steps:

a   Use the Available Components drop-down list to select **Registry**.

b   In the New Component Name text box type an instance name for the new registry scan.

c   Click **Add + Edit**.

d   Use the Edit instance dialog to modify the attributes, as necessary. The PROPERTY, CNDITION and DEPTH attributes define the hive, registry subkey and depth of the scan, respectively.

e   Click **OK** to save your changes.

The registry scan instance is automatically created and attached to the audit package.

4   Connect the audit package to an audit service.

5   Entitle the audit service for the registry scan to the appropriate machines or users.

6   The registry scan service is deployed during the first connection to an entitled agent. Upon the next connection, the registry scan inventory is collected and passed to the CM Messaging Server, which posts it to the ODBC database for Inventory.

7   View the Registry Scan report from the CM Reporting Server.

Updating the CM-CSDB for Windows Registry Scans

To be able to scan the Registry of Windows machines using WMI, HP includes the Registry class in the AUDIT Domain of the CM Configuration Server Database. If your existing database does not include the Registry class, see the following import procedure to add the class to your database.

The import deck is available as enhancement fix: ER3201044, available from HP Technical Support.

Figure 4   REGISTRY Class added to the AUDIT Domain.
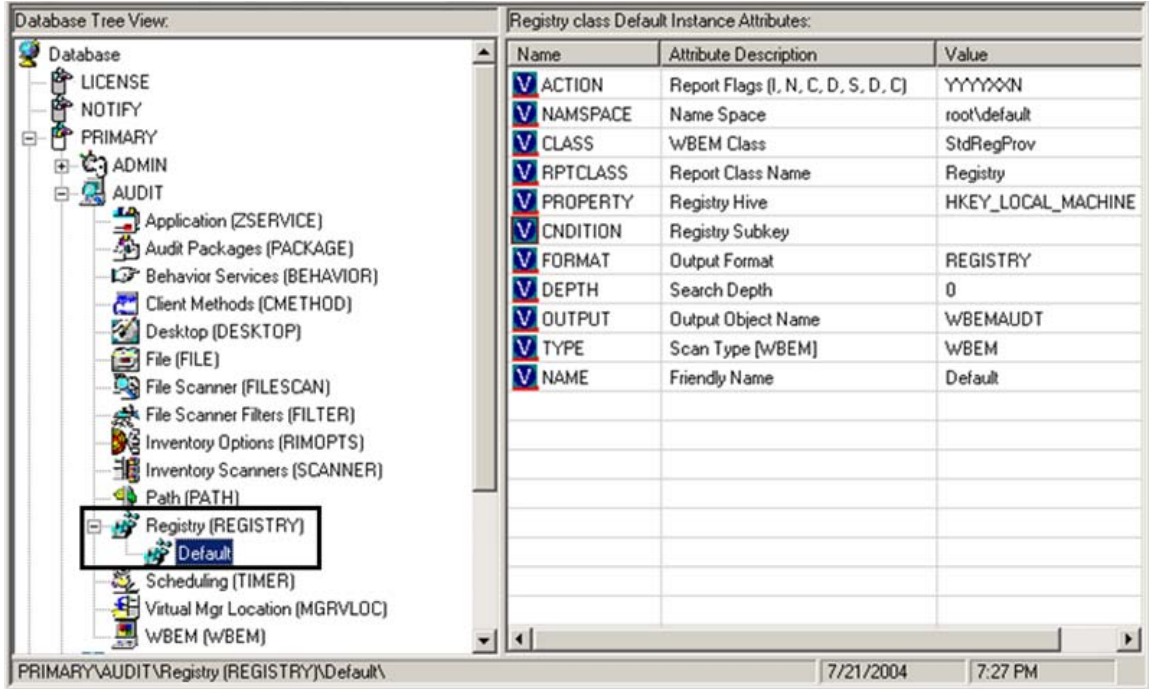
To install the CM-CSDB update for the AUDIT Registry class

⚠   Do not use this procedure to add a class already on your CM-CSDB.

1   Stop the CM Configuration Server.

2   From the downloaded installation media, go to the `db_import` directory. This folder contains the import deck for the new class to be added to the CM-CSDB.

AUDIT.Registry (**REGISTRY**)   REGISTRY.XPC

3   Copy the `REGISTRY.XPC file` into the CM Configuration Server `bin` subdirectory.

The default location of the bin directory is: `Drive:\Program Files\ Hewlett-Packard\CM\ConfigurationServer\bin`.

4   Open a command prompt and navigate to the CM Configuration Server's `bin` directory.

5   Execute the following command from the bin directory to import the AUDIT.REGISTRY class to your database.

> ⚠️ Do not import classes already on your database.

`ZEDMAMS VERB=IMPORT_CLASS,FILE=REGISTRY.XPC,PREVIEW=NO`

A return code of 0 indicates that there were no errors reported during the import, and the updates are applied to the database. If there were errors, you can type **`zedmams.log`** to open and view that log.

6   Restart the CM Configuration Server.

The Registry Class is newly available in the AUDIT Domain, as shown in Figure 4 on page 81.

Inventory Database Tables

The inventory reporting database includes the tables shown in Figure 5 below, among others.

Figure 5   Standard Inventory Database – Tables.

Create table in Design view	rWin32_DisplayConf	rWin32_Process
Create table by using wizard	rWin32_DisplayControllerConf	rWin32_Processor
Create table by entering data	rWin32_DMAChannel	rWin32_Product
AppEvent	rWin32_Environment	rWin32_SerialPort
DeviceConfig	rWin32_FloppyController	rWin32_Service
DeviceErrors	rWin32_FloppyDrive	rWin32_Share
DeviceMap	rWin32_Group	rWin32_SoftwareElement
DeviceNotify	rWin32_IDEController	rWin32_SoftwareFeature
DeviceServices	rWin32_IRQResource	rWin32_SoundDevice
DeviceState	rWin32_Keyboard	rWin32_StartupCommand
DeviceStatus	rWin32_LoadOrderGroup	rWin32_SystemDriver
FileAudit	rWin32_LogicalDisk	rWin32_SystemEnclosure
HAppEvent	rWin32_LogicalMemoryConf	rWin32_TimeZone
HDeviceErrors	rWin32_LogicalProgramGroup	rWin32_USBController
HDeviceState	rWin32_MemoryArray	rWin32_UserAccount
HDeviceStatus	rWin32_MemoryDevice	rWin32_VideoController
rCIM_Product	rWin32_MotherboardDevice	
rWin32_BIOS	rWin32_NetworkAdapter	
rWin32_BootConf	rWin32_NetworkAdapterConf	
rWin32_Bus	rWin32_NetworkConnection	
rWin32_CacheMemory	rWin32_OperatingSystem	
rWin32_CDROMDrive	rWin32_PageFile	
rWin32_ComputerSystem	rWin32_PageFileSetting	
rWin32_ComputerSystemProduct	rWin32_PageFileUsage	
rWin32_Desktop	rWin32_ParallelPort	
rWin32_DesktopMonitor	rWin32_PnPEntity	
rWin32_DeviceMemoryAddress	rWin32_PointingDevice	
rWin32_DiskDrive	rWin32_PortResource	
rWin32_DiskPartition	rWin32_Printer	

The table names denote the origin of the data that they contain. For example, the **rWin_LogicalMemoryConf** table will be populated with data from the Win32_LogicalMemoryConfiguration WBEM class.

Tables that begin with rWin32_ are populated with the data from WBEM queries. Tables that do not start with rWin32_ are populated with data from non-WEBM sources.

The recommended product for viewing Inventory is the CM Reporting Server. Refer to the *CM Reporting Server Guide* for more information.

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.

- Prior to beginning any tasks using the CM Inventory Manager, you must enable the drag-and-drop functionality for the newly created RIMOPTS Class instances.

- Use the CM Reporting Server for viewing the information obtained by auditing agent computers.

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.

This guide is provided to assist you with installing and implementing the CM Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

Auditing Types

When configuring your audits, it is beneficial for the administrator to understand exactly what types of things can be audited and what the expected results from an audit will be.

The CM Inventory Manager allows for three types of audits:

- File auditing

- WBEM auditing

- Hardware auditing

File Auditing

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.

See What Happened during the Scan on page 115 for additional information on the RIMFSCAN and the RIMDIFF methods.

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

Table 14   AUDIT.FILE Class Instances

Attribute	Description and Examples
SCANFOR	Indicate a fully qualified path and file name to search for. Wildcards are permitted. *Drive*:\WinNt**.dll
ACTION	The RIMDIFF method performs actions on the files discovered on the user's computer during the Agent Connect.

| Attribute	Description and Examples
	• **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 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.

Attribute	Description and Examples
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 additional information for a file. • Set the value to **0** to not collect additional 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. • (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.

Attribute	Description and Examples
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. *SystemDrive*:\Data\&(ZOBJPID)\&(name)
ZRSCMMEM	PDS member name. This field is optional.
PRODUCT	The product name. See ZVERINFO on page 90 for more detail.
PRODVERS	The product version. See ZVERINFO on page 90 for more detail.
ORGNAME	The organization name. See ZVERINFO on page 90 for more detail.
INTERNAL	The internal data element encoded in the file. See ZVERINFO on page 90 for more detail.
VERSION	The version of the file. See ZVERINFO on page 90 for more detail.
LANGUAGE	The language of the file. See ZVERINFO on page 90 for more detail.
VENDOR	The product vendor. See ZVERINFO on page 90 for more detail.
ZRSCCRC	Resource CRC.

Attribute	Description and Examples
ZCRCINFO	Collect file CRC. [Y/N] Default is **N**. • Set the value to **Y** to collect CRC information for a file. • Set the value to **N** to not collect CRC information. • If blank, defaults to **N**. Caution: Collecting file CRC information can dramatically extend the time it takes to collect information on the target machine.
ZMD5INFO	Collect file MD5 information. [Y/N] MD5 information is a 32-character value that can be used to uniquely identify a file based on its content. Default is **N**. • Set the value to **Y** to collect MD5 information for a file. • Set the value to **N** to not collect MD5 information. • If blank, defaults to **N**. Caution: Collecting MD5 information can dramatically extend the time it takes to collect information on the target machine.
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.

Use the CM Admin Agent Explorer to view the FILEAUDT object results as shown in Figure 6 on page 93.

Figure 6   FILEAUDT object.
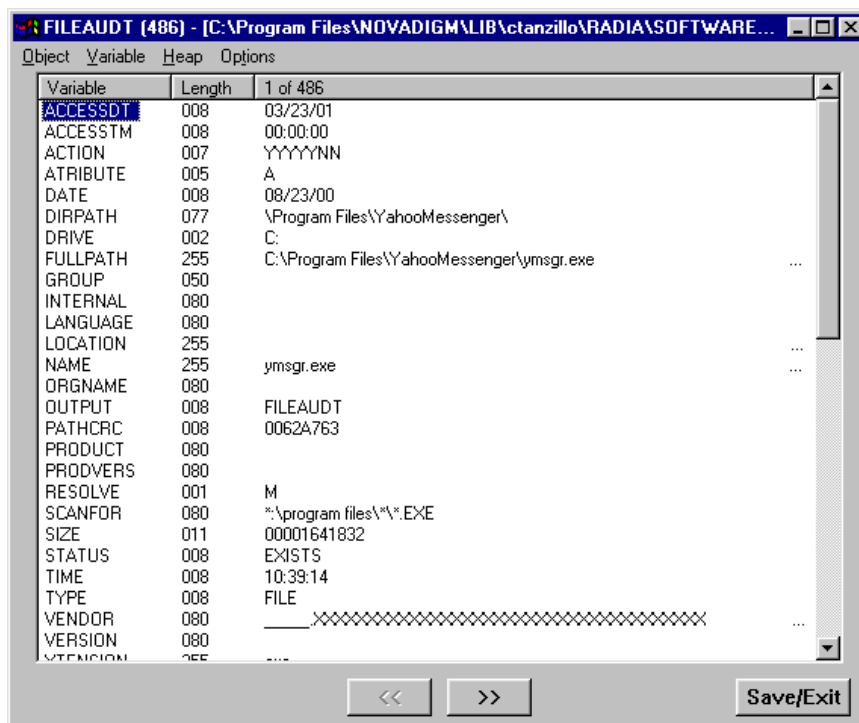
The FILEAUDT object contains one heap for each file discovered during the scan for the audit service. It contains the attributes from the AUDIT.FILE class instance that controlled the scan, as described above. It also contains the following attributes:

Table 15   FILEAUDT Object

Attribute	Description
ACCESSDT	The date of the most recent access of this file.
ACCESSTM	The time of the most recent access of this file.
ATRIBUTE	A string listing the attributes of the file: R = Read only A = Archive S = System H = Hidden C = Compressed

Attribute	Description
DATE	The date of the most recent modification to this file.
DIRPATH	The directory path of the file.
DRIVE	The system drive location of the file.
FULLPATH	Fully qualified path and file name of the file.
PATHCRC	A unique number that indicates the CRC path used for differencing.
RESOLVE	The value of M indicates that the CM Configuration Server resolves each heap of the FILEAUDT object individually. This value cannot be modified.
SIZE	File size in bytes.
STATUS	Indicates the status of the file on the agent computer. Possible values are: • **Exists** This is the first time scanning for this file and it was found. • **New** This file was added to the file system of the agent computer since the last scan was performed. • **Update** This file exists in the new and previous scans. There have been changes to the date, time, size, and/or version. • **Deleted** This file was present in the previous scan but is missing in the new scan. • **Not found** No files were found that matched this request.
TIME	The time of the most recent modification to this file.
XTENSION	The file extension. This is useful for sorting and querying back-end database tables that store the data found in this object.

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.

⚠️ CM Inventory Manager leverages Microsoft's Windows Management Instrumentation (WMI) to collect hardware and software inventory data by using WMI queries. Some WMI queries can traverse the network contacting other servers in the enterprise to collect the requested information. This may result in large volumes of data being returned, and could have a significantly negative effect on network performance. An example of this would be querying all users on the network using the W32_UserAccount WMI class. Extreme caution must be taken to understand the scope of these queries to ensure unexpected results do not occur. While CM Inventory Manager provides an interface to WMI and its providers, it cannot control how these queries are satisfied. It is the customer's responsibility to safeguard against using WMI queries that span the network, if this behavior is not desired.

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

Figure 7   AUDIT.WBEM class instances.
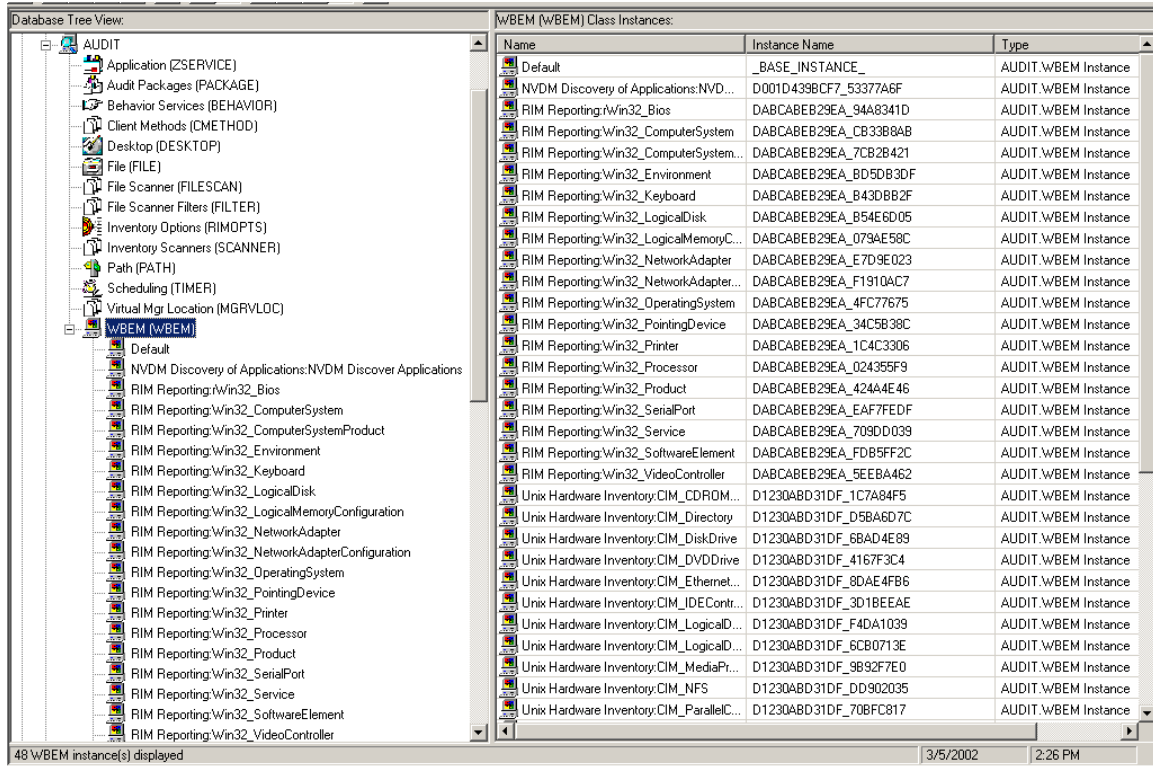
Table 16 below describes the attributes of the AUDIT.WBEM instance.

Table 16   AUDIT.WBEM Instance

Attribute Name	Description
ACTION	The 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,

Attribute Name	Description
	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.
	• **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	The name of the WBEM namespace to query or HARDWARE.
CLASS	The 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.

Attribute Name	Description
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's 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 Inventory Manager agent stores the results of a WBEM scan in a WBEM object. This object can be found in the service node of the agent object tree. The results are also sent to the CM Configuration Server.

In addition to the attributes described in Table 16 on page 96, the WBEM object also contains the following:

Table 17   WBEM Object Attributes in the Agent

Attribute	Description
ZOBJCID	Object child ID.
ZOBJCLAS	The targeted class for the audit such as ZRSOURCE or ZSERVICE.
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.

Attribute	Description
ZOBJRSIZ	The resource size maintained by the CM Configuration Server.
ZOBJTIME	The latest time under the current node.
ZRSCSRC	The name of the program promoted the resource.

WBEM Object Processing

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

1 At the end of the agent connect, the ZTASKEND rexx method on the CM Configuration Server is called and creates commands to invoke the QMSG executable.

2 QMSG.EXE places the WBEMAUDT objects into the CM Configuration Server \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 CM 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 After it is posted to the CM 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 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-CSDB:

- 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.

Hardware Auditing

Each time a CM agent connects to the CM Configuration Server, information about the agent'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 as follows:

Figure 8   ZCONFIG object.

A separate ZCONFIG object is calculated and stored for each service installed or updated during the Agent Connect. To force the transfer of the hardware information, the ZCONFIG attribute *must* be set to Y in the POLICY.USER class.

Figure 9   POLICY.USER Class – ZCONFIG attribute

The ZCONFIG object contains a wealth of information about the agent computer's hardware.

Figure 10   ZCONFIG object.

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 additional information to be stored.

The following table describes the attributes that are stored in the ZCONFIG object.

Table 18   ZCONFIG Object

Attribute	Description
GATEWAY	Router for your subnet.
HALCOMP	Company of HAL.DLL

Attribute	Description
HALDATE	Date and time of HAL.DLL
HALFNAME	Original name of HAL.DLL
HALFVER	Internal version of HAL.DLL
HALINAME	Name of HAL.DLL
HALLANG	Language of HAL.DLL
HALPNAME	Product name of HAL.DLL
HALPVER	Product version of HAL.DLL
HALSIZE	Size of HAL.DLL
IPADDR##	IP address of network adapter (there can be multiple addresses).
LADAPT##	Network card (there can be multiple network cards).
REBOOTD	Last re-boot date.
REBOOTT	Last re-boot time.
SUBNET##	Subnet mask.
ZGATEWAY	Looks at GATEWAY attribute.
ZHDWBIOS	BIOS type.
ZHDWCDDR	CM agent's CD-ROM drive letter.
ZHDWCOMP	Computer name.
ZHDWCPU	Current CPU type.
ZHDWFPU	Current FPU type.
ZHDWIPAD	The IP address of the computer.
ZHDWKYBD	Keyboard type.
ZHDWLANA	LAN adapter.
ZHDWLANG	Language setting.
ZHDWMEM	Total physical memory (RAM).
ZHDWMEMF	Total free memory (RAM).
ZHDWMOUS	Mouse type.

Attribute	Description
ZHDWNET#	Network card information (can be multiple cards).
ZHDWNNET	Number of network cards.
ZHDWOS	Computer's operating system and version.
ZHDWOSCL	Operation system classification (Workstation or Server)
ZHDWOSDB	Operating system's build number.
ZHDWOSOG	Organization.
ZHDWOSOW	Owner.
ZHDWOSSR	Windows 9x Sub-Version Number (i.e., A, B, C)
ZHDWPA##	Printer information.
ZHDWPPAR	Number of parallel ports.
ZHDWPPRN	Number of printers available.
ZHDWPSER	Number of serial ports.
ZHDWVIDO	Video type.
ZHDWVMSI	MSI Version
ZHDWVRES	Video resolution.
ZHDWXPAG	Page size.
ZHWCPU01	CPU type.
ZHWFPU01	FPU type.
ZMODEM	Modem present? Y or N
ZOBJDATE	The date of the CM agent connect for this service.
ZOBJNAME	HARDWARE_SCAN (hard coded).
ZOBJTIME	The time of the CM Agent Connect.
ZSUBNET	The subnet mask.
ZUSERID	The name of the user who connected.

Whenever a CM agent connects to the CM Configuration Server, certain hardware information concerning the subscriber is automatically forwarded to the CM Inventory Manager ODBC database as part of the CM Messaging Server processing of CORE objects. The hardware information is viewable through the CM Reporting Server.

Summary

- The CM Inventory Manager allows for file auditing, WBEM auditing, and hardware auditing.

- The RIMFSCAN and the 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 a CM 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 attribute *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 pre-packaged Audit Applications (ZSERVICE).
- Know how to design your own Audit Packages (PACKAGE).

This manual is provided to assist you with installing and implementing the CM Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

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 as follows:

Figure 11   Sample Auditing services.
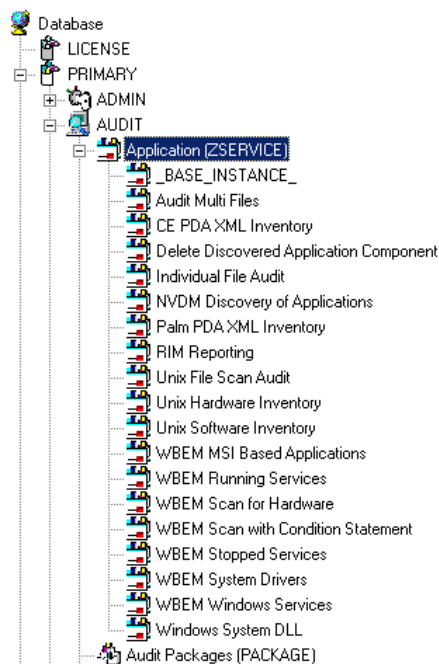
These sample services represent common scenarios for inventory collection and management. The best way to develop your own audit services is to study the samples that were installed with the CM Inventory Manager upgrade.

The sample audit services are described in the following table:

Table 19   Sample of Auditing Services

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

For documentation purposes, we will configure the sample audit service Audit Multi Files. The file type we will be auditing is indicated in the SCANFOR attribute within the instance. This instance directs the CM Inventory Manager agent to scan for any `*:\program files\*\*.exe` files

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 12   SCANFOR attribute of the Audit Multi Files instance.
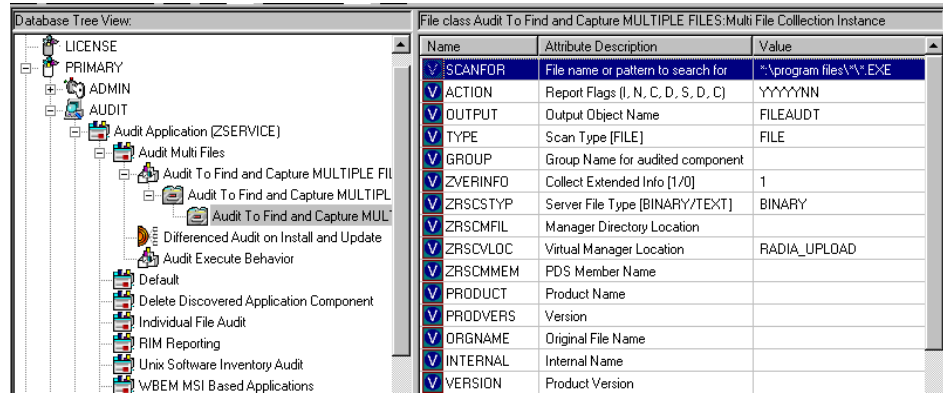
To configure a sample Audit package

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 on Application (ZSERVICE) to expand the class.
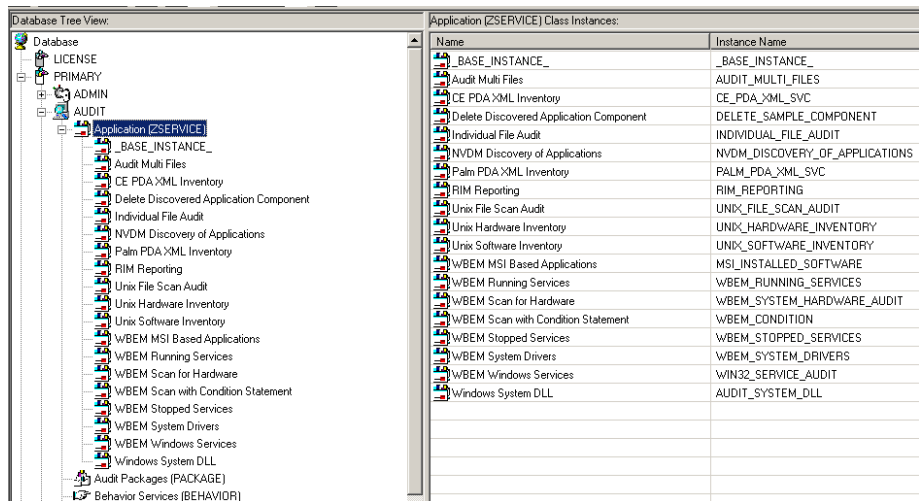
4   Scroll to and expand the POLICY Domain.

For our example, we would like all users that are members of the Workgroup class to select this audit package from their CM Application Self-service Manager.

5   Expand the POLICY.WORKGROUPS class.
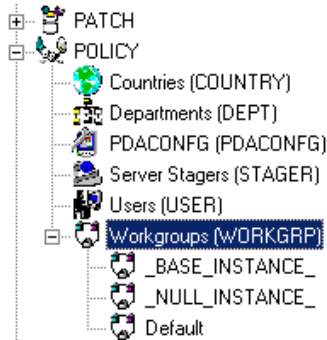
6   Select the **Audit Multi Files** package from the **ZSERVICE** class and drag it to the POLICY.WORKGROUPS class and drop it on the **Default** instance.

The Select Connections Attribute window opens.
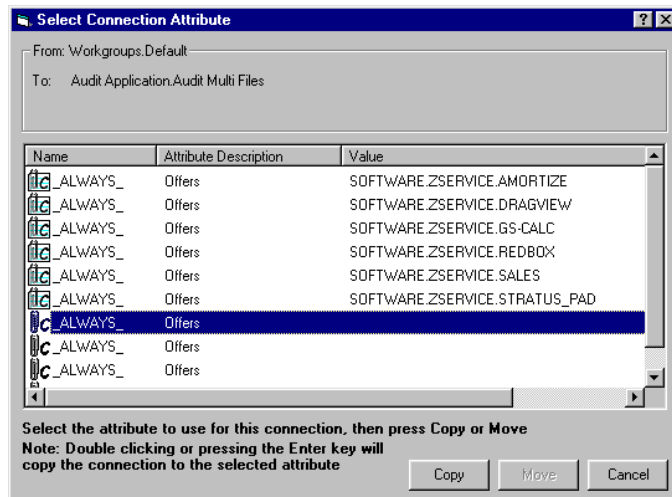
7   Click **Copy** to add this package.

The Confirm Connection dialog box opens.

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

The Audit Multi Files package is added to WORKGRP Class.
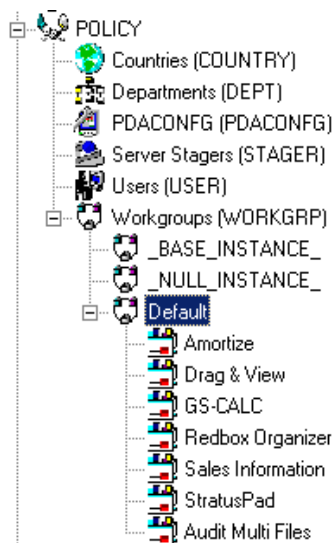
The collection of inventory information occurs on the CM Inventory Manager agent computer when a user connects to the CM Configuration Server as follows:

- Through a CM Application Self-service Manager agent connect, when the user launches that program.

  or

- Through the CM Application Manager agent when the user double-clicks the CM Connect icon on his desktop, or is scheduled or notified to connect.

Figure 13 below shows the available Audit Multi Files package that a CM Application Self-service Manager user would see when connecting to the CM Configuration Server:

Figure 13   CM Application Self-service Manager shows Audit Multi Files

When the subscriber selects and installs the Audit Multi Files package from the CM Application Self-service Manager, there are really two connections. 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.

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

What Happened during the Scan?

Use the CM Admin Agent Explorer to locate the ZSERVICE for the **Audit Multi Files** package in the LIB directory.

Figure 14   AUDIT_MULTI_FILES in the LIB directory.
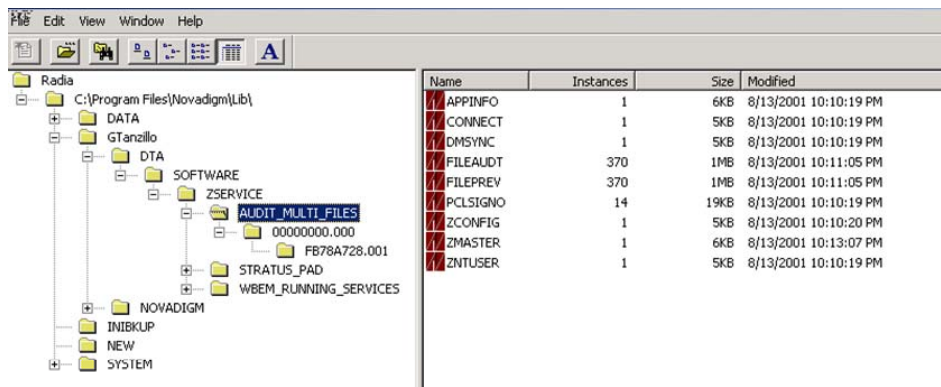
Within the ZSERVICE, note the two objects, **FILEAUDT** and **FILEPREV**. These objects are created and stored in the ZSERVICE of the LIB directory whenever an audit package is installed. The FILEAUDT object contains one heap for each file discovered during the auditing scan. It also contains the attributes from the AUDIT.FILE instance that controlled the scan.

The AUDIT.FILE class instances in an audit package control the auditing 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.

Figure 15   Auditing with the RIMFSCAN and RIMDIFF methods.
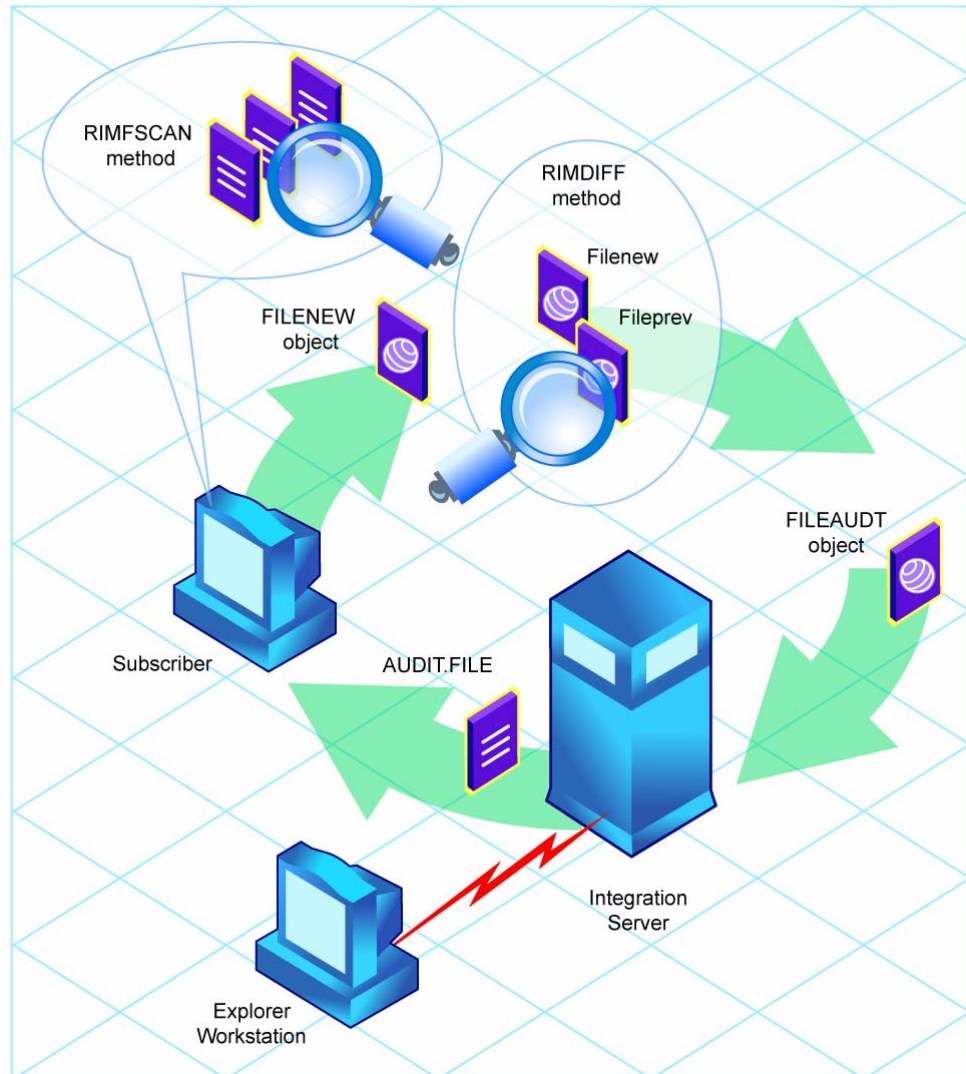
- The RIMFSCAN method scans the CM agent's file system based on the values in the AUDIT.FILE class instance in the audit package. It constructs an object called FILENEW. The FILENEW object contains one heap per file discovered during the current scan.

- The RIMDIFF method compares scan results from the current scan (the scan done during the current Agent Connect stored in the FILENEW object) with scan results from a previous scan (the scan done during a previous Agent Connect process stored in the FILEPREV object). It will

construct the FILEAUDT object that is then sent to the CM Configuration Server. The RIMDIFF method then deletes the FILEPREV object and will rename the FILENEW object to FILEPREV.

For our particular example, there were 486 instances for both the FILEAUDT and the FILEPREV object located on the CM agent'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.

- Inventory information is collected on the CM Inventory Manager 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.

- The RIMFSCAN method scans the CM agent's computer file system based upon the values contained in the AUDIT.FILE class instance in the audit package. It constructs an object called FILENEW.

- The RIMDIFF method compares the scan results from the current scan (the scan done during the current CM agent connect process stored in the FILENEW object) with the scan results from a previous scan (the scan done during a previous Agent Connect process stored in the FILEPREV object). It constructs the FILEAUDT object that is then sent to the CM Configuration Server.

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.

This manual is provided to assist you with installing and implementing the CM Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

Audit Packages or PACKAGE Class

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

By expanding the Audit Packages (PACKAGE) class, you will see the audit package instances.

Figure 16   Audit Package (PACKAGE) class.
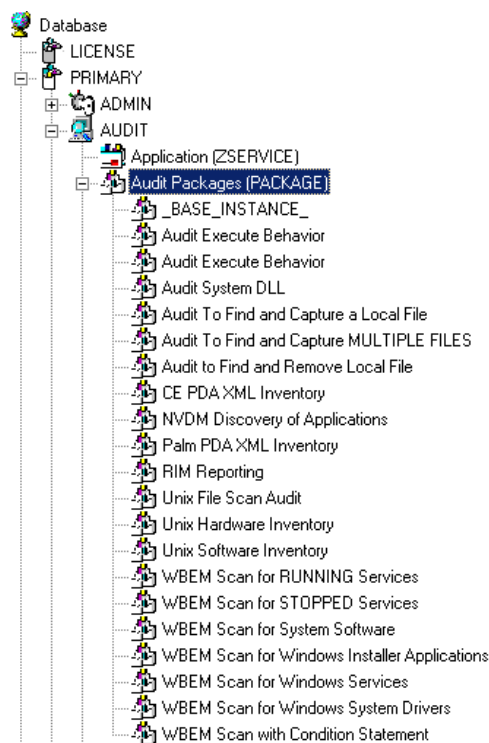
A complete audit service consists of several connected instances in the AUDIT Domain. The audit package instance is a container that "owns" the instances connected to it. For example, open the AUDIT.ZSERVICE class and double-click on the Individual File Audit instance.

Figure 17   Individual File Audit instance.

In the example, the Individual File Audit ZSERVICE instance "owns" the Audit to Find and Capture a Local File instance. The fact that a package instance owns a component class instance means that all of the instances are managed as a package unit. If the package instance is deleted, all of its owned class instances are automatically deleted as well.

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

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

Table 20   Inventory Options (RIMOPTS) Class

Instance	Description
Default	Contains the base instance attributes for the RIMOPTS Class. • Collect attribute is set to Diff. • Runexec attribute is set to IU. • ZSVCTYPE attribute is set to I.

Instance	Description
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.
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.

See Chapter 4, The AUDIT Domain for additional information about RIMOPTS attributes.

Finally, a connection to an auditing behavior is needed.

Figure 18   Connection to an Audit Behavior
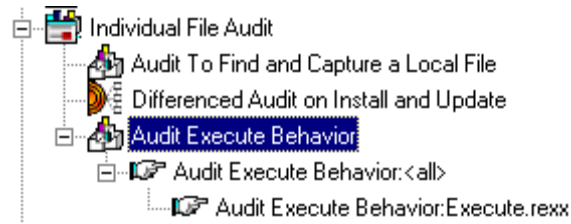
The audit behavior owned by the Individual File Audit ZSERVICE is connected to the Behavior Services (BEHAVIOR) class within the AUDIT Domain.

The BEHAVIOR class in the AUDIT Domain remains unchanged from the BEHAVIOR class within the SOFTWARE Domain. Refer to the *CM Configuration Server Database Reference Guide, Chapter 8: The SOFTWARE Domain* for the description of the attributes found within this class.

Using the CSDB Editor Create/Maintain Audit Services

By using the CSDB Editor, we will walk through the construction of a file audit. An instance of the AUDIT Domain's Audit Package (PACKAGE) Class contains information about the inventory information to collect, and what action to take with that collected information.

Prior to beginning the creations 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? (See Configuring Timers for Audit Collection on page 135 for information concerning timers.)

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

⚠ If you are creating a WBEM Audit Package, be aware CM Inventory Manager leverages Microsoft's Windows Management Instrumentation (WMI) to collect hardware and software inventory data by using WMI queries. Some WMI queries can traverse the network contacting other servers in the enterprise to collect the requested information. This may result in large volumes of data being returned, and could have a significantly negative effect on network performance. An example of this would be querying all users on the network using the W32_UserAccount WMI class. Extreme caution must be taken to understand the scope of these queries to ensure unexpected results do not occur. While CM Inventory Manager provides an interface to WMI and its providers, it cannot control how these queries are satisfied. It is the customer's responsibility to safeguard against using WMI queries that span the network, if this behavior is not desired.

To create a new Audit package

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

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

> ▶ The User ID, as shipped from HP, is `RAD_MAST`. No password is necessary. This may have been changed during installation. Check with your Configuration Management security administrator to obtain your own User ID and Password, if necessary.

3 Double-click **PRIMARY**.

4 Expand the **AUDIT Domain**.

5 Double-click on **Audit Packages (PACKAGE)** class.
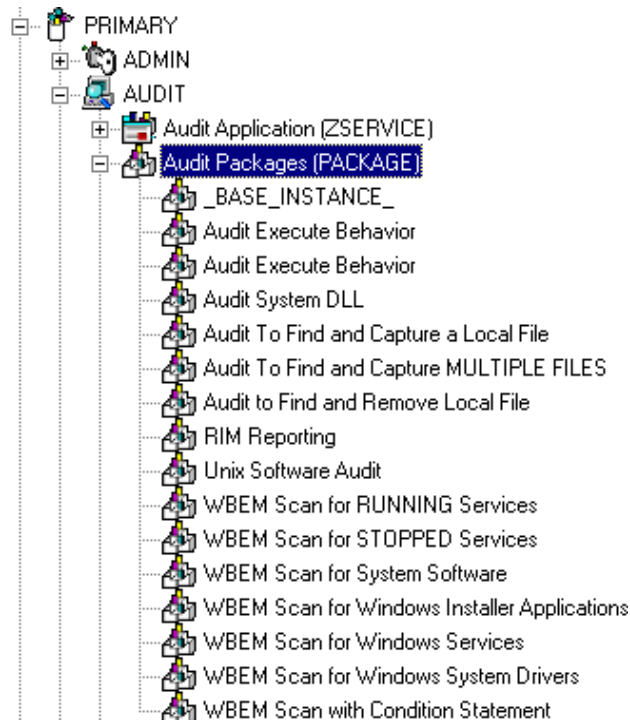
As an example, we will create a new auditing package called Log Finder. This package will scan a user's computer for .log files, capture them, and return the results to the administrator.

6   Right-click on the **Audit Packages (PACKAGE)** Class.

A shortcut menu opens.
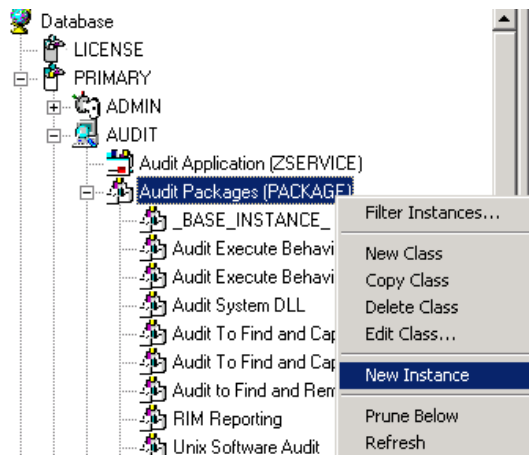
7   Select **New Instance** from the shortcut menu.

    The Create Instance dialog box opens.

8   Enter a new display name for the package instance. This friendly name
    will appear in the tree view.

9   Enter a name for the Create a new Audit Packages (PACKAGE) instance
    name. 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 Log Finder package is added to the AUDIT.PACKAGE class.
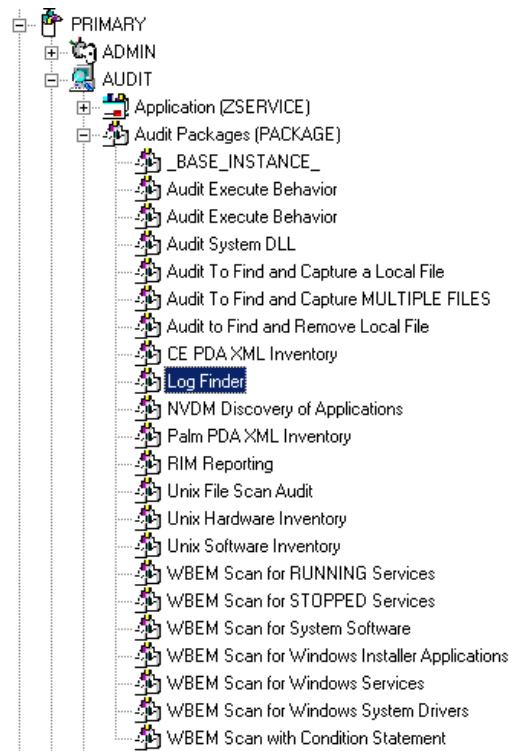
Once the Log Finder package is created, you will need to add its components.

To add a component to an Audit package

1   Right-click on the Log Finder package.

2   A shortcut menu opens.
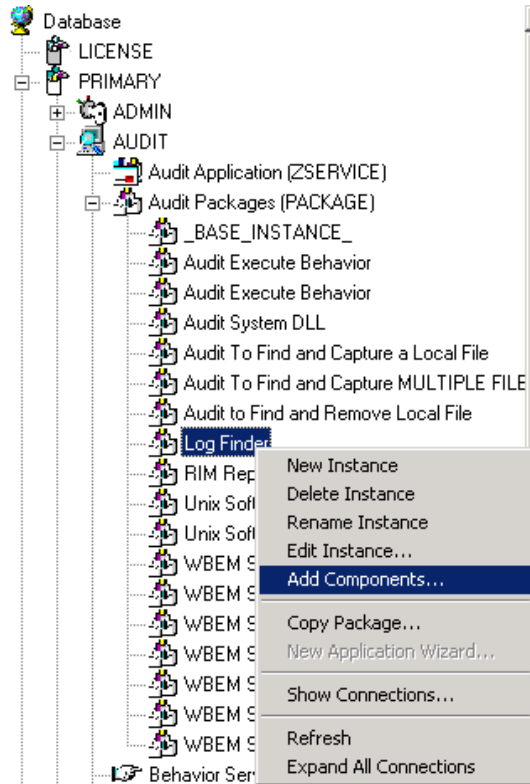
3   Select **Add Component** from the shortcut menu.

The Add Components dialog box opens.

4   Click on the **Available Components** drop-down arrow, and select **File**.
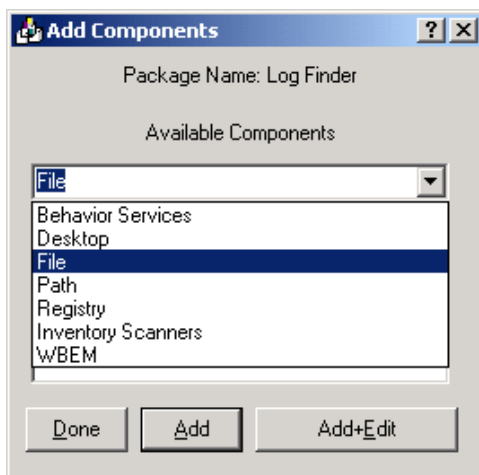
5 In the **New Component Name** text box, enter the new component name.
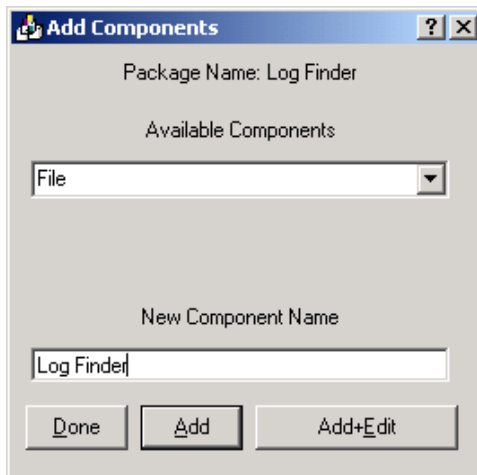
6 Click **Add+Edit**. The component is added to the package and the Editing
  Instance dialog box opens.
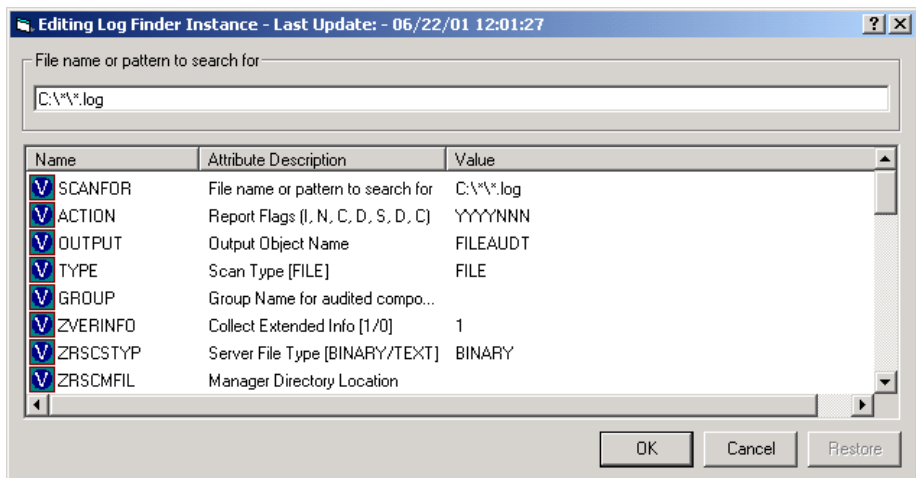
In the Editing Instance dialog box, you can edit the instances that will be
used in your audit.

> Use the AUDIT.FILE class instances to help you decide which
> instances you may want to edit.

For our example, we changed the SCANFOR attribute to `C:\*\*.log`.

Continue to edit, line-by-line, as necessary.

7 Click **OK** when you are done with your edit.

8   Click **Yes** to save your changes.

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

▶   While working in 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 in the tree view.

2   Right-click **Audit Application (ZSERVICE)** and a shortcut menu opens.
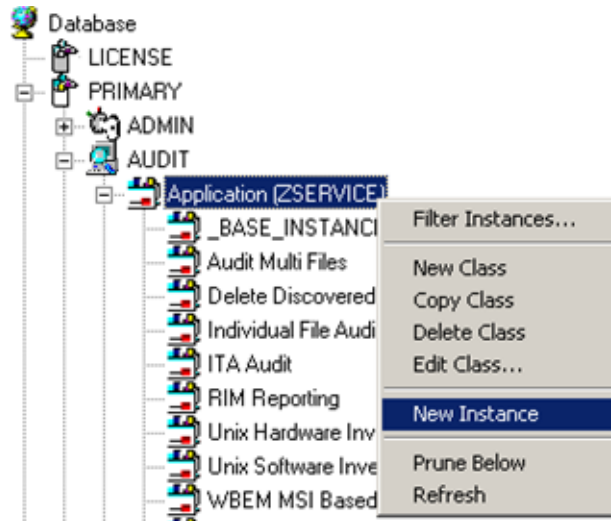
3   Select **New Instance** from the shortcut menu.

The Create Instance dialog box opens.

4   Type a display and an instance name.

5   Click **OK**. The ZSERVICE Log Finder is added to the AUDIT.ZSERVICE class.
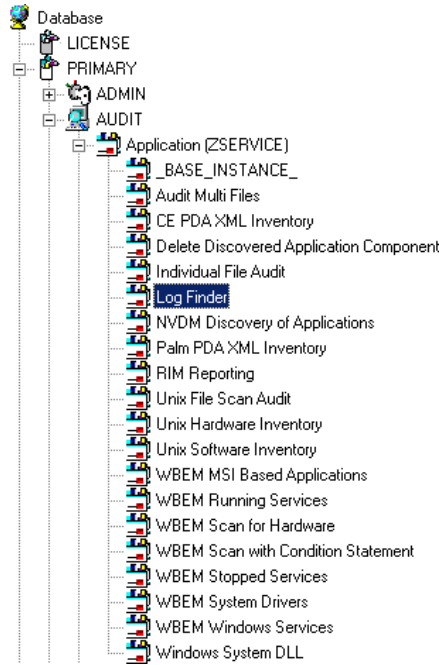
6 Use the CM Admin CSDB Editor to connect the Log Finder package to
   the Log Finder service.
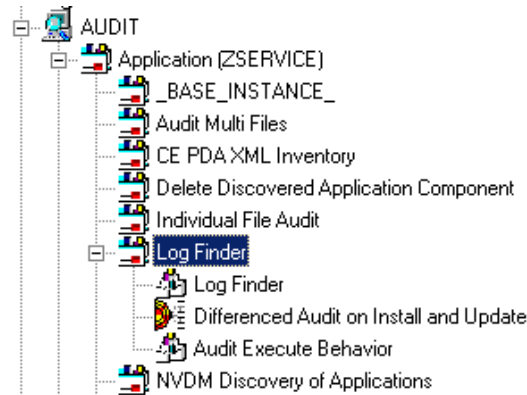
Once the connection to the ZSERVICE has been completed, various optional
steps can be taken.

You might want to ask yourself the following questions:

- Will the service appear in the CM Application Self-service Manager?
  Should the ZSVCNAME be changed? Should I enter additional
  information that may appear in the CM Application Self-service
  Manager?

- Will this be a mandatory or optional service?

- Will the service have a certain length of time to be active?

- Do I want to confirm if the service is installed or not?

The answers to these questions can help you decide how to customize the service.

For our example, we wanted to change the service name from Unknown to Log Finder. We also wanted to make this service available to users in the CM Application Self-service Manager, so we have changed the ZSVCMO attribute from mandatory to mandatory *and* optional. We would like the CM Configuration Server to report back and store any .log files that are found. Therefore, we will change the ZRSCMFIL attribute to capture and store this information on the CM Configuration Server's directory.

Figure 19   Log Finder ZSERVICE attributes

Name	Attribute Description	Value
ZSTOP000	Expression Resolution Method	
ZSTOP001	Expression Resolution Method - 001	
ZSTOP002	Expression Resolution Method - 002	
ZSTOP999	Stop Unless Radia Connect	
ZSVCNAME	Service Name/Description	Log Finder
ZSVCTTYP	Application Target Type [A/S]	
ZSVCMO	Mandatory or Optional Service [M/O]	MO
ZSVCCSTA	Service Status on Client (999)	999
ZSVCPRI	Service Create Ordering [01-99]	
ALWAYS	Contains	AUDIT.PACKAGE.LOG_FINDER
ALWAYS	Contains	
ALWAYS	Contains	
ALWAYS	Contains	
ALWAYS	Contains	
ALWAYS	Contains	AUDIT.RIMOPTS.DIFF_INSTALL_UPD...
ALWAYS	Contains	AUDIT.PACKAGE.AUDIT_EXECUTE_B...
ALWAYS	Utility Resolution Method	
ZCREATE	Service Installation Method	
ZINIT	Service Initialization Method	
ZDELETE	Service Delete Method	
ZUPDATE	Service Update Method	
ZVERIFY	Service Verify Method	
ZREPAIR	Service Repair Method	
ZAVIS	Available,Verified,Installed,Sync F	YXNX
PUBDATE	Published Date of Service	
VERDATE	Verfied Date of Service	
UPGDATE	When Application was Upgraded on De	
UPDDATE	Upgrade Date (Programmatic)	
INSTDATE	Installed Date	
DELDATE	Delete Date	
AUTHOR	Author Name	
DESCRIPT	Application Description	

Use the CM Admin CSDB Editor to connect and deploy the Log Finder audit service.

In this particular example, the user sees the new audit service in the CM Application Self-service Manager.

Figure 20   Log Finder in the Application Self-service Manager.

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, system 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 in 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.

This guide helps you install and implement the CM Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

The Scheduling (TIMER) Class

The Scheduling (TIMER) class enables the CM administrator to set a timer on the CM agent computer that 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 pre-determined schedule.

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

Housed within the AUDIT.Scheduling (TIMER) class are 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.

- **Discover Installed Software Timer**
  executes a ZSERVICE weekly between 8:30 am and 10:30 pm. Use this particular timer in conjunction with the ZSERVICE Discovery of Applications that audits the ADD/REMOVE PROGRAM part of the OS.

These sample packages can be copied and modified, changing the time parameters to suit your needs. Refer to the *CM Administrator User Guide* for information on copying an instance. Or, you can create a new timer instance by following the instructions given in Creating a Timer Instance on page 141.

Figure 21   AUDIT Scheduling (TIMER) class
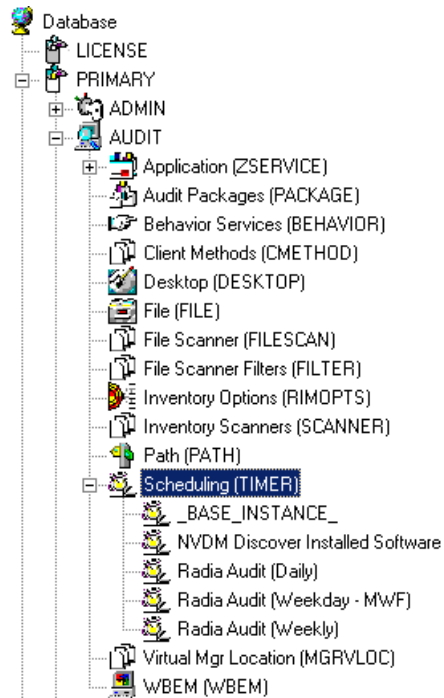
Timers can be set to expire periodically (hourly, daily, weekly, monthly, or at defined intervals), on a specific date, or at a specific time. Each CM agent is installed with the HP OVCM Scheduler service. This service contains an executable timer component that executes any program on the end-user desktop when a timer expires.

Typically, the HP OVCM 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 contains descriptions of the Scheduling (TIMER) class attributes:

Table 21   Scheduling (TIMER) Class

Attribute	Description
ZOBJPRI	Sets the priority for deployment of the ZTIMEQ object. The ZTIMEQ object is deployed relative to the other elements being deployed during the CM Agent Connect. The 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. We recommend you leave the default configuration of USER.
ZSCHDEF	Indicates when the timer expires. The syntax varies depending on the frequency of expiration that can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, and 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 A.M. (non-peak) scheduled event that is sent to the CM 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 4:00:00, the command in the instance will be executed the same day at 4:00:00. • After 4:00:00, the command in the instance will be executed immediately. If ZSCHTYPE = DEFERRED and it is:

Attribute	Description
	• Before 4:00:00, the command in the instance will be executed the *next* day at 4:00:00.
	• After 4:00:00, the command in the instance will be executed the *next* day at 4:00:00.
	Example 2:
	Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,04:00:00)
	If ZSCHTYPE = IMMEDIATE and it is:
	• Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00.
	• Friday and after 4:00:00, the command in the instance will be executed immediately.
	If ZSCHTYPE = DEFERRED and it is:
	• Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00.
	• Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00.
ZSCHFREQ	This attribute 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	This attribute 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	Friendly name for this instance.

Attribute	Description
APPSVC	Application.
REQUEST	Application request.
DOMAIN	Server's domain name.
IPADDR	Server's IP address/name.
SOCKET	Server's socket number.
MGRNAME	Server's name.
ZCREATE	Scheduler CREATE method that runs on the CM agent computer. This value is inherited from the base instance and should not be changed.
ZVERIFY	Scheduler VERIFY method that runs on the CM agent computer. This value is inherited from the base instance and should not be changed.
ZUPDATE	Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZDELETE	Scheduler DELETE method that runs on the CM agent computer. This value is inherited from the base instance and should not be changed.
RUNSYNC	Sets the value of Yes or No for the synchronous timer execution. The default value is Yes.
ZNOPING	Controls the automatic sensing of a network connection between the CM 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.

Attribute	Description
	Use this attribute when there is a possibility that the CM agent will not be able to connect with the CM Configuration Server. This attribute is especially useful for mobile users. Note: In order to use this attribute, you must add it to the TIMER class template.
PINGDLAY	Sets the amount of time between pings in milliseconds. The default setting is 2000 milliseconds.
PINGCNT	Sets the number of ping attempts to be made by the CM Configuration Server. The default setting is 3.

This section describes 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?

Creating a Timer Instance

To create a timer in the CM-CSDB, use the CS 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 of the Scheduling (TIMER) classes and can be connected to an Application (ZSERVICE) class instance in either the SOFTWARE or AUDIT Domains.

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

For additional information concerning the Scheduling (TIMER) class, see the *Deploying Applications* chapter in the *CM Application Manager and Application Self-Service Manager Guide.*

1 From the Start menu, go to **Programs → HP OVCM Administrator → CM Admin CSDB Editor**. The Security Information dialog box opens.

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

> The User ID, as shipped from HP, is **RAD_MAST**., with no password necessary. This may have been changed during installation. Check with your CM security administrator to obtain your own User ID and Password, if necessary.

3 Double-click **PRIMARY**.

4 Double-click **AUDIT**.

5 Right-click **Scheduling (TIMER)**.
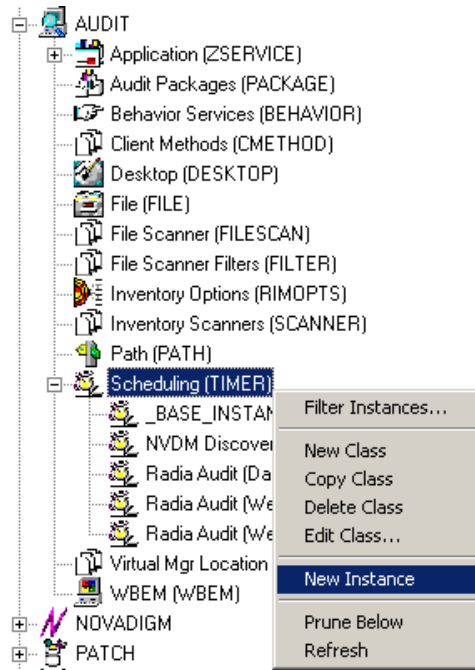
6 Select **New Instance**. The Create Instance dialog box opens.

7 Type a name for the new timer instance, such as Log Audit Timer.

8 Click **OK**. The timer instance appears in the Scheduling (TIMER) Class.
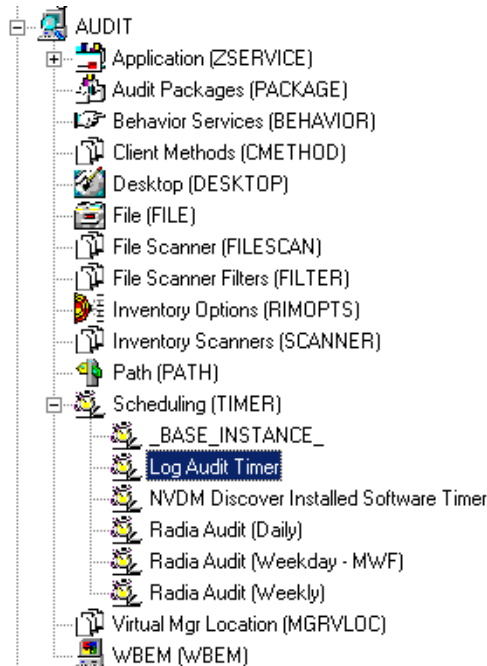
Specifying Timer Settings

Whether you have copied an existing timer or you have created a new Timer instance, you need to review and/or customize your timer settings. Refer to the *CM Application Manager and Application Self-Service Manager Guide* for more information on how to specify the CM agent timer settings.

Specifying ZSCHDEF

Use the ZSCHDEF attribute to define the time interval and date and time to execute the command line. The syntax varies depending upon the interval chosen. When configuring ZSCHDEF, the attribute is set in the following form depending on the interval.

```
DAILY(<DATE>,<TIME>[,<LIMIT>])

HOURLY(<DATE>,<TIME>[,<LIMIT>])

WEEKLY(<DATE>,<TIME>[,<LIMIT>])
```

```
WEEKDAY(<DAY of Week>,<TIME>[,<LIMIT>])

NUMDAYS(<DATE>,<TIME>[,<LIMIT>],<Number of Days>)

INTERVAL(<DATE>,<TIME>[,<LIMIT>],<Number of Seconds>)
```

> In the case of NUMDAYS and Interval, the Optional parameter
> <LIMIT> is between mandatory parameters. If the optional
> parameter is omitted the place must be held with a double comma.
> Example:
> NUMDAYS: NUMDAYS(20000803,08:00:00,12:00:00,14)
> NUMDAYS: NUMDAYS(20000803,08:00:00,,14)

- 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:

    YYYYMMDD

- 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 values for time and limit_time are optional. They are specified in
  the following form:

    HH:MM:SS

- The value for count is optional. It is specified as an integer.

The timer expiration can also be configured on the value of ZSCHFREQ. Use
Table 21 on page 138 to help you determine the appropriate syntax.

Table 22   Syntax of ZSCHDEF Attributes

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 seven days at 1:00 am.
WEEKDAY	WEEKDAY(Name of Weekday*,01:00:00)	Every *Name of Weekday** at 1:00 AM. The weekday must be specified in uppercase.

Type	Syntax	Timer Expires
HOURLY	HOURLY(&ZSYSDATE,08:41:00)	Hourly starting at 8:41 AM on the system's 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.

Specifying ZSCHTYPE

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

- **IMMEDIATE**
  will execute the command specified in the ZRSCCMDL attribute immediately if the date and time indicated in the ZSCHDEF attribute 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 the ZSCHFREQ to specify whether the timer should expire once (ONCE) or repeatedly (PERIODIC) according to the frequency specified in ZSCHDEF.

Specifying ZRSCCMDL

Use the 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 uid=&(ZMASTER.ZUSERID),startdir=&(ZMASTER.LOCALUID),
mname=&(ZMASTER.ZMGRNAME),dname=&(ZMASTER.ZDOMNAME),sname=&(ZSERV
ICE.ZOBJNAME)
```

> Execution causes CM to launch the AUDIT service behavior,
> (EXECUTE.REXX) attached to the AUDIT service.
>
> The parameters indicated in the radskman command may differ
> depending upon customer specific implementations.

Specifying ZNOPING, PINGDLAY, and PINGCNT

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

Refer to the *CM Application Manager and Application Self-service Manager
Guide* for more information about the ZNOPING attribute.

- 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 the ZRSCCMDL attribute is executed. The PENDING attribute in
    the CM agent's ZTIMEQ object is then set to N. This indicates that
    the Scheduler service does not need to ping the CM Configuration
    Server again.

  — Set ZNOPING to **W** if you are specifying an end limit in the ZCHDEF
    attribute. The Scheduler pings the CM Configuration Server before
    executing the command. If the CM Configuration Server is
    unavailable, the ZPENDING flag is set to "W". If the ZSCHEDEF has
    a limit time, then when that time passes, the ZPENDING flag is set
    to N, and the Scheduler will not attempt to execute the command
    until its next scheduled time.

  — If the CM Configuration Server is not pinged successfully, the timer
    is not processed any further. The ZPENDING attribute value remains
    set to Y. The next time the Scheduler service expires, it should ping
    the CM Configuration Server again.

If ZNOPING is set to N, also use the PINGDLAY and PINGCNT attributes to further specify the timing and number of pings between the agent computer and the CM Configuration Server.

- If ZNOPING is set to N, PINGDLAY specifies the time in milliseconds between pings. The default is 2000.

- If ZNOPING is set to N, PINGCNT specifies number of ping attempts. The default is 3 attempts.

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 **Log Audit Timer** to the **Log Finder** ZSERVICE created earlier in this document.

Figure 22: Log Audit Timer instance connected to the Log Finder service.
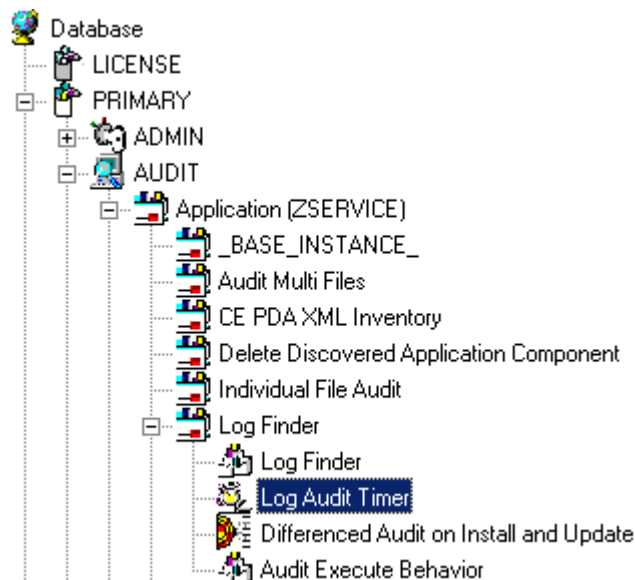
Then connect the AUDIT.ZSERVICE.Log Finder to a user or group of users in the POLICY Domain.

Figure 23: Log Finder attached to a user

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, File Scan, or an MSI request. The audit service definition may also be installed at a time when an audit scan is not desirable. For example, when an end user visits the Application Self-service Manager and mandatory applications are processed as defined in the embed tag enterprisemanagement=auto.

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

• Customizing the Inventory Options (RIMOPTS) attribute.

    and

• Updating the embed tags in the html file for the 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 PVCM Administrator→ CM Admin CSDB Editor**. The CM Admin CSDB Editor Security Information dialog box opens.

> The User ID, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. 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   Expand the **PRIMARY File** and the **AUDIT Domain**.
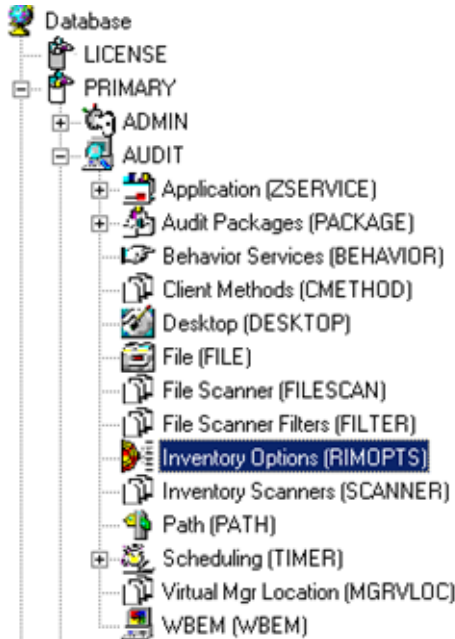
4   Create a new instance in the **Inventory Options (RIMOPTS)** class called CM_AUDIT_NO_EXECUTE, and click **OK**. The Create Instance dialog box opens.

Next, you will need to edit the CM Audit No Execute instance.

5   Expand the **Inventory Options (RIMOPTS)** class and double-click the **CM Audit No Execute** instance.
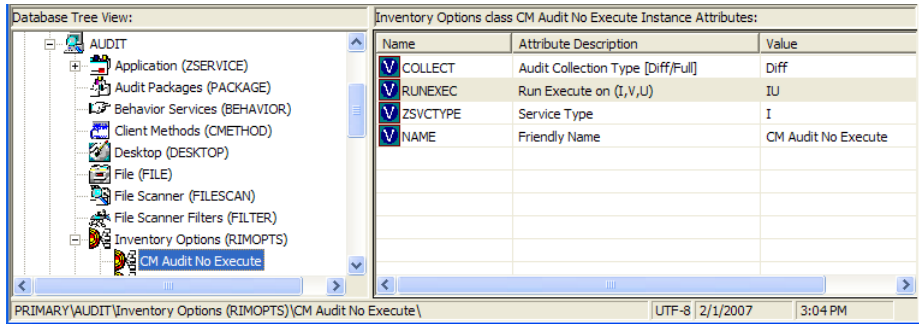
6   Double-click the **RUNEXEC** attribute in the list view to edit it. Remove any attribute information. This will ensure that the audit service will not run during the installation, verification, or update function.
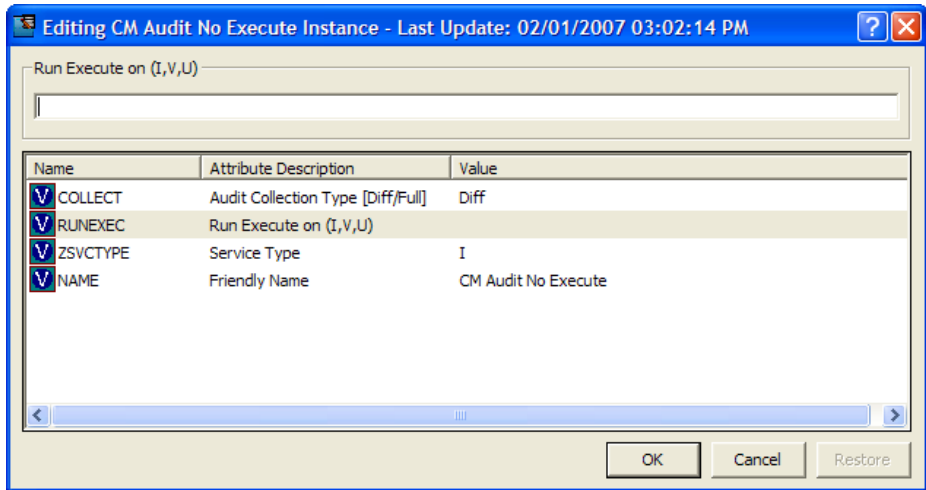
Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the RIM_REPORTING service.

7   Right-click on **RIM_REPORTING** Service in the AUDIT class.

8   Select **Edit Instance**.
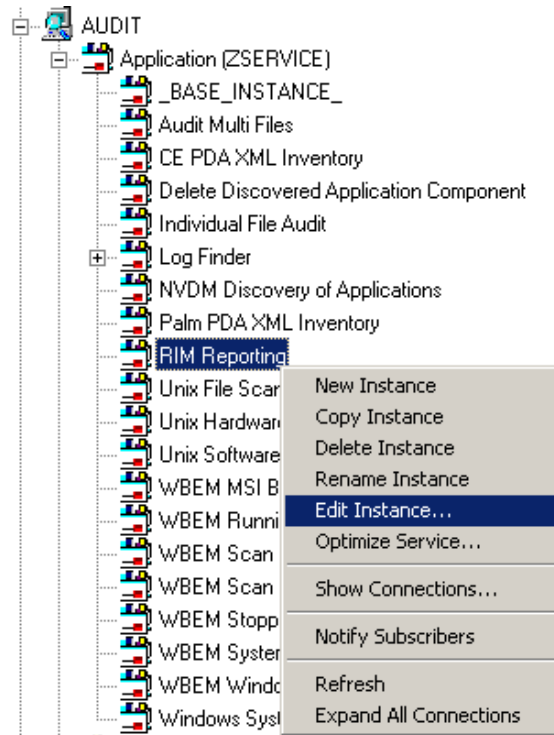
9 Locate the _ALWAYS_ Contains attribute with the value of AUDIT.RIMOPTS.DIFF_INSTALL_UPDATE and change it to a value of **AUDIT.RIMOPTS.CM_AUDIT_NO_EXECUTE**.
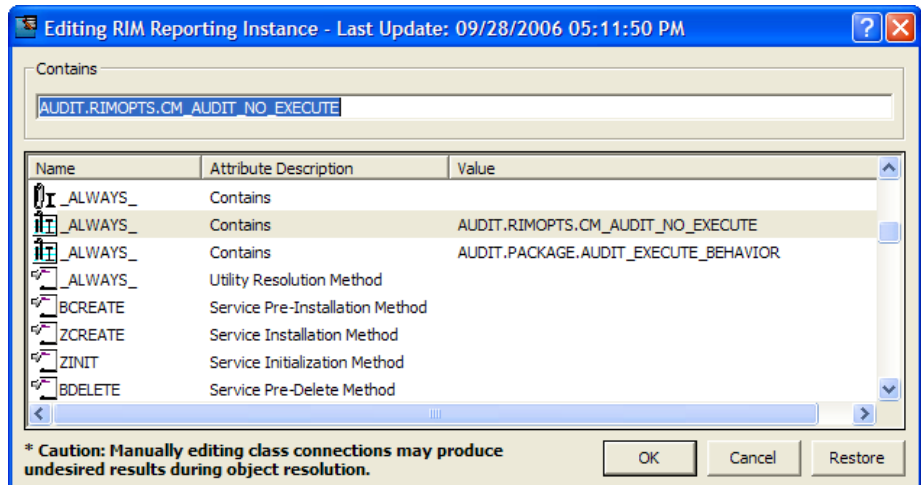
10 Next, to define the audit service as Mandatory, locate the ZSVCMO field and set it to M. This will cause the initial TIMER definition associated with the audit service to be created on the CM agent.
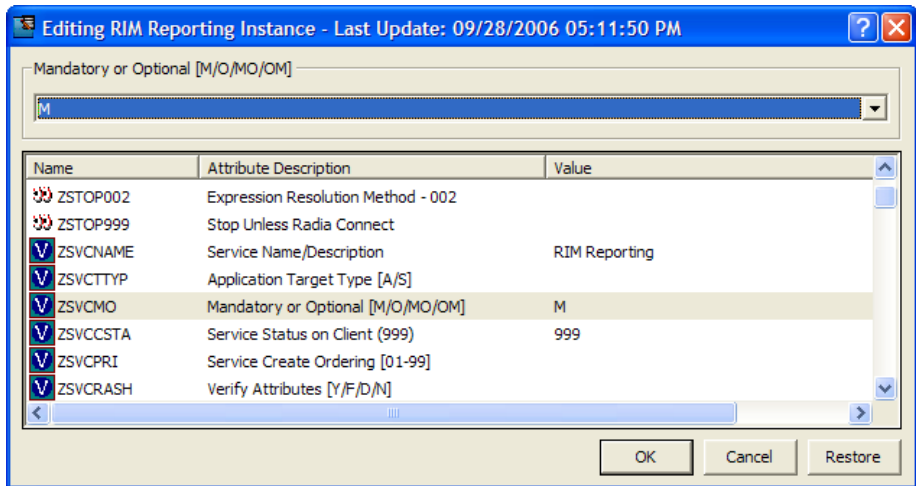
The CM Audit No Execute instance is now connected to the RIM Reporting service.
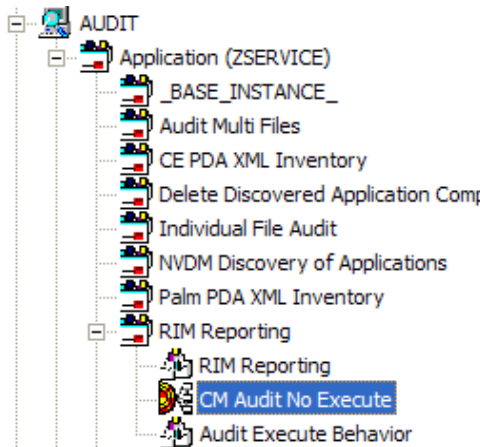
Summary

- The Scheduling (TIMER) class enables the CM administrator to set a timer on the CM agent computer. The timer will cause one or more audit services to be processed whenever the time 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 HP OVCM Scheduler service lies dormant in the background, and wakes up once per minute to see if a timer has expired.

- Use the ZSCHDEF to indicate when the timer should expire.

- Use the 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 client computers.

- Be able to navigate through the information collected by clicking on 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 additional 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 Management Reports:

- Executive Summaries
- Operational Reports
- Hardware Reports

- Software Reports
- Readiness Reports

Figure 24   Inventory Management reports

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

Table 23   Hardware Reporting Views

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

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

Table 24   Software Reporting Views

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

Windows Vista Readiness Reports

Use the Display Options to show Windows Vista readiness reports. These reports contain information you can use to determine individual device readiness for an upgrade to Windows Vista. The CM Reporting Server determines Vista readiness based on the following criteria:

- CPU Speed

- Memory

- System Drive Total

- System Drive Free

Refer to Microsoft's support web site for additional Vista readiness information.

To display Windows Vista Readiness reports

1   In the Display Options area, select **Inventory Management Reports**.

2   Select **Readiness Reports**.

3   Select **Windows Vista**.

4   View the reports and charts available to determine the Windows Vistas upgrade readiness of your devices. The Readiness Status and Additional Information columns contain information about the current level of readiness for each device.

Refer to the *CM Reporting Server Guide* for more information.

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 25   Inventory Management Related Data Filters

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

Some filters only allow a text entry. Others have a Show available options button or magnifying glass to open a filter lookup window.

To get help when entering filter, point to the help icon and a tooltip specifies the syntax and gives examples.
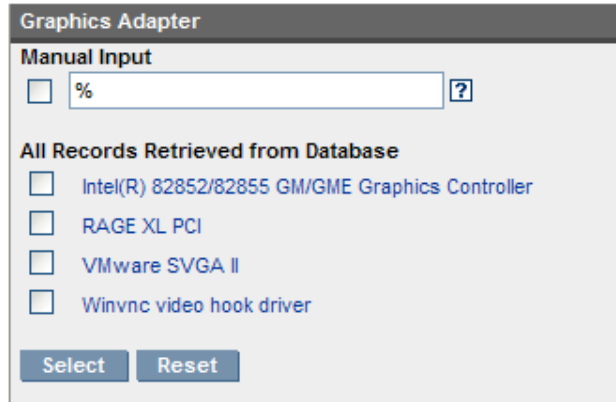
Figure 26   Expand a filter
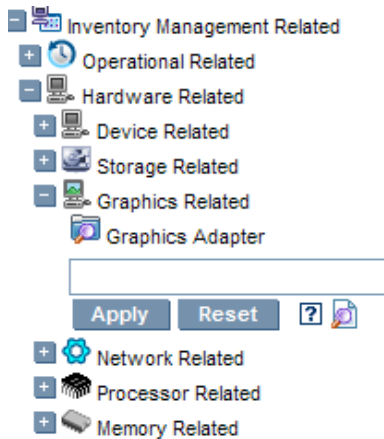
Click the magnifying glass to open the filter lookup window.

Figure 27   Select the filter.

Click any of the available criteria check boxes to select the criteria you would like to use in your filter. For additional 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 Detail and Summary Reporting Tables

Table 25   Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
Applications	Managed Applications	device_id/Subscriber	AppEvent
		service_id/Service	
		ctime/Created	
		mtime/Modified	
		app_name/Application Name	
		event/Event	
		del_time/Date Deleted	
		ver_time/Date Verified	
		inst_time/Date Installed	
		fix_time/Date Fixed	
	Audited Applications	Cim-show-apps.tsp	
	Installed Applications	Installed-apps.tsp	
	Add/Remove Applications	Installed-uninstalled-apps.tsp	
WBEM Applications	Installed Products	wName/Tag	rCIM_Product
		wVendor/Vendor	
		wVersion/Version	
		wIdentifyingNumber/Software Spec	
		wCaption/Caption	
	Installed Filesets /	wPartComponent/Fileset	rCIM_SoftwareFeatureElements
	Packages	wName/Tab	rCIM_SoftwareElement
		wVersion/Version	
		wSoftwareElementID/SoftwareSpec	
		wTargetOperatingSystem/TargetOS	
		wManufacturer/Vendor	
		wCaption/Caption	
		wInstallDate/Install Date	
	Audited Applications	wCaption/Application Name	rNVD_Product
		mtime/Modified	

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		CIM_Product.wDescription/type	
		wName/Name	
		wVendor/Vendor	
		wVersion/Version	
		wInstallState/Installed	
		wInstallDate/Date Installed	
Audited Files	Audited Files	name/Name	FileAudit
		version/Version	
		status/Status	
		vendor/Vendor	
		product/Product	
		prodvers/Product Version	
		scanfor/Scanned	
		file_date/File Date	
		file_size/File Size	
		mtime/Modified	
		file_type/File Type	
		path/Path	
Configuration Summary for Windows	O/S Configuration	mtime/Modified	rWin32_OperatingSystem
		_wOS/OS	
		wRegisteredUser/Registered User	
		wOrganization/Organization	
		wSerialNumber/S/N	
		wSystemDirectory/Sys Dir	
		WtotalPageFileSpace/PageFileSize (mb)	
	Hardware	manufacturerr/Manufacturer	rWin32_ComputerSystemProduct
		_model/Model	RWin32_systemEnclosure
		_wSNTag/S/N	RWin32_Processor
		wManufacturer,wCurrentClockSpeed/Processor	RWin32_LogicalMemoryConf
		wTotalPhysicalMemory/Physical Memory (MB)	rWin32_Computer System
		wSystemType/System	rWin32_Bios
		_wBios / Bios	
		_wKybd/keyboard	RWimd32_Keyboard
		_wMouse/Mouse	rWin32_PointingDevice

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		_wVideo/Video/Video	rWin32_VideoController
		_wDriverName/Printer	rWin32_Printer
		_WSerialPort/Serial Ports	rWin32_SerialPort
		_wParallelPort/Parallel Ports	rWin32_ParallelPort
	Network Adapter information	Wbem-show-network.tsp	
	Disk Drive Information	Wbem-show-drives.tsp	
	Environment	Wbem-show-environment.tsp	
	Windows Services	Wbem-show-services.tsp	
	Device Configuration	Ctime/Created	Device Config
		Mtime/modified	
		Os/OS	
		Os_level/ OS Level	
		Sysdrv/Sys Drive	
		Sysdrv_total/ Sys Drive Size (MB)	
		Sysdrv_free/Sys Drive Free (MB)	
	Software (AGENT and	Person/Person	Device Config
		Organization/Organization	
		Language/Language	
		Protocol/Protocol	
		Timeout/Timeout	
		Trace/Trace	
		Edmsys/Sys Dir	
		Edmlib/Lib Dir	
		Edmlog/Log Dir	
	Hardware	Ipaddr/ IP Address	Device Config
		Macaddr/ MAC Address	
		Bios/ Bios	
		Cpu/CPU	
		Memory/Mem (MB)	
		Keyboard/Keyboard	
		Mouse/Mouse	
		Video/Video	
		N_serial/Serial(#)	
		N_Parallel/Parallel (#)	

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		N_PRINTER/printer (#)	
WBEM Features	Audited Features	wProductName/Product Name	RWind32_SoftwareFeature
		mtime/Modified	
		WInstallDate/Date Installed	
		wVendor/Vendor	
		wVersion/Version	
Installed Applications	Audited Files	Show-fileaudit.tsp	
	Installed Applications	WFileDescription / Application Name	RNVD_Installed_Apps
		Mtime / Modified	
		WPath / Path	
		WoriginalFileName / Executable	
		WFileVersion / Executable Version	
		WcompanyName ? Vendor	
		WProductName / Product Name	
		WProducttVersion / Version	
	Add/Remove Applications	Installed-uninstall-apps.tsp	
WBEM Elements	Audited Elements	mtime/Modified	rWin32_SoftwareElement
		wName/Name	
		wVersion/Version	
		wInstallDate/Date Installed	
		wManufacturer/Manufacturer	
		wPath/Path	
PDA Devices	PDA Devices	mtime/Modified	rNVD_PDASystem
		wName/Name	
		wDescription/Type	
		wStatus/Status	
WBEM PDA Config	Configuration	Mtime/Modified	rCIM_OperatingSystem
		wCaption, wVersion / OS	
		wFreePhysicalMemory/Free Physical Memory (MB)	
		wTotalVirtualMemorySize/Total Virtual Memory (MB)	
		wFreeVirtualMemorySize/Free Virtual Memory (MB)	
	PDA Installed Products	Show-pda-inst-prod.tsp	RCIM_OperatingStystem

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
wbem-show-environment.tsp	Environment	mtime / Modified	rWin32_Environment
		wUserName / Account	
		WSystemVariable / System Variable	
		wName / Name	
		wAttributeValue / Value	
wbem-show-services.tsp	Window Services	mtime / Modified	rWin32_Services
		wDisplayName / Services	
		wState / Status	
		wStartMode / Startup	
		wName / Name	
		wStartName / Logon	
		wDesktopInteract / Interact with Desktop	
		wPathName / Path	
Wbem-show-network.tsp	Network Adapter Information	Mtime / Modified	RWin32_NetworkAdapterConf
		Wdescription / Type	
		WIPAddress / IP Address	
		WMACAddress / MAC Address	
Wbem-show-drives.tsp	Disk Drive Information	Mtime / Modified	RWin32_LogicalDisk
		WDeviceID / Drive Letter	
		WDescription/Type	
		WfileSystem / File System	
		WSize / Size (MB)	
		WFreeSpace / Free Space (MB)	
		WProviderName / Provider Name	
		WvolumneSerialNumber / Serial Number	
Installed-uninstall-apps.tsp	Add/Remove Applications	WDisplayName / Application Name	RNVD_Installed_Uninstall
		Mtime / Modified	
		WUninstallString / Uninstall String	
Show-pda-inst-prod.tsp	PDA Installed Products	Mtime / Modified	Rnvd_Product
		Wdescription/ Type	
		WStatus / Status	
		WVersion / Version	

Table 26   General Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
Show-Config	Device Configuration	ctime/Created	DeviceConfig
		mtime/Modified	
		os/OS	
		os_level/OS Level	
		sysdrv/Sys Drive	
		sysdrv_total/Sys Drive Size (MB)	
		sysdrv_free/Sys Drive Free (MB)	
	Software	person/Person	
		organization/Organization	
		language/Language	
		protocol/Protocol	
		timeout/Timeout	
		trace/Trace	
		edmsys/Sys Dir	
		edmlib/Lib Dir	
		edmlog/Log Dir	
	Hardware	ipaddr/IP Address	
		macaddr/MAC Address	
		bios/Bios	
		cpu/CPU	
		memory/Mem (MB)	
		keyboard/Keyboard	
		mouse/Mouse	
		video/Video	
		n_serial/Serial (#)	
		n_parallel/Parallel (#)	
		n_printer/Printer (#)	
Status - Application Events	Application Events	device_id/Subscriber	AppEvent
		service_id/Services	
		ctime/Created	
		mtime/Modified	
		app_name/Application Name	
		event/Event	
		status/Status	

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		del_time/Date Deleted	
		ver_time/Date Verified	
		inst_time/Date Installed	
		fix_time/Date Fixed	
Status - Connect	Connect Status	mtime/Modified	DeviceStatus
		duration/Duration	
		mrc/Return Code	
		reason/Reason	
		svc_count/Services (#)	
		rsrc_count/Files (#)	
		rsrc_transfer/Files Tx (#)	
		rsrc_transfer_size/Files Tx (Sz)	
		ctime/Created	
	Errors	mtime/Modified	DeviceErrors
		type/Type	
		code/Code	
		reason/Reason	
		module/Module	
		object/Object	
		component/Component	
Status - Services	Service State	mtime/Modified	DeviceServices
		serviceid/Service	
		svc_actv/Svc Actv	
		rsrc_active/Files Active (#)	
		rsrc_inactive/Files Inactive (#)	
		ver_error/Vers Err	
		reason/Reason	
Status - Notify	Notification Status	device_id/Subscriber	DeviceNotify
		nfy_status/Status	
		mtime/Modified	
		nfy_reason/Reason	
		nfy_cmd/Command	
		ctime/Created	
		nfy_type/CommsType	
		nfy_attempts/Attempts (#)	
		nfy_userid/User Id	

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		nfy_addr/Address	
		nfy_port/Port	
		nfy_maxretry/Max (#)	
		nfy_delay/Delay (s)	
		nfy_timeout/Timeout (s)	
		nfy_retry2/Retry2 (#)	
		nfy_retry2/Retry2 (#)	
		nfy_timeout2/Timeout2 (s)	
Status - Summary	Connect Status	mtime/Modified	DeviceState
		mrc/Return Code	
		duration/Duration	
		svc_count/Services (#)	
		rsrc_count/Files (#)	
		reason/Reason	
	Agent State	mtime/Modified	
		state/State	
		svc_count/Services (#)	
		rsrc_count/Files (#)	
		rsrc_error/File Err	
		ver_error/Vers Err	
		reason/Reason	
	Service State	Status-services.tsp	
Status - Detailed	Connect Status	*same as* Status Connect *entries*	
	Agent State	Mtime / Modified	
		State / State	
		Svc_count / Services (#)	
		Rsrc_count / Files (#)	
		Rsrc_error / File Err	
		Ver_error / Vers Err	
		Reason / Reason	
	Service State	Status-services.tsp	
	Errors	*same as* Status Connect *entries*	

Table 27   History Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
Application Events	Application Event History	device_id/Subscriber	HAppEvent
		service_id/Service	
		mtime/Modified	
		app_name/Application Name	
		event/Event	
		status/Status	
		del_time/Date Deleted	
		ver_time/Date Verified	
		inst_time/Date Installed	
		fix_time/Date Fixed	
		nvd_domain/Domain	
		nvd_class/Class	
Connect	Connect History	mtime/Modified	HDeviceStatus
		duration/Duration	
		mrc/Return Code	
		reason/Reason	
		svc_count/Services (#)	
		rsrc_count/Files (#)	
		rsrc_transfer/Files Tx (#)	
		rsrc_transfer_size/Files Tx (Sz)	
Errors	Error History	mtime/Modified	HDeviceErrors
		type/Type	
		code/Code	
		reason/Reason	
		module/Module	
		object/Object	
State	State History	mtime/Modified	HDeviceState
		state/State	
		svc_count/Services (#)	
		ver_error/Vers Error	
		rsrc_count/Files (#)	
		rsrc_error/File Err	
		rsrc_active/Files Active (#)	
		rsrc_active_size/Files Active (Sz)	

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		rsrc_inactive/Files Inactive (#)	
		rsrc_inactive_size/Files Inactive (Sz)	
		reason/Reason	

Table 28   Summary Reporting

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
Show - Subscribers	Application Subscribers	device_id/subscriber	DeviceStatus
		mtime/Modified	AppEvent
		llength [*]/InstalledApps (#)	
Show - Applications	Applications	app_name or service_id / Application Name	AppEvent
		count (device_id)/Subscribers	DeviceServices
Show - System Drivespace	Subscribers System Drive Space	device_id/Subscriber	DeviceConfig
		sysdrv/Sys Drive	
		sysdrv_total/Sys Drive Size (MB)	
		sysdrv_free/Sys Drive Free (MB)	
		(sysdrv_free*100)/sysdrv_total / Percent Free	
Show - IP Addresses	Subscribers IP Addresses	device_id/Subscriber	DeviceConfig
		ipaddr/IP Address	
		macaddr/MAC Address	
WBEM Configuration	Configuration	userid/Subscriber	rWin32_Bios
		mtime/Modified	rWin32_OperatingSystem
		wCaption, wBuildNumber, wCSDVersion/OS	rWin32_LogicalDisk
		wSystemDirectory/System Drive	rWin32_ComputerSystem
		wSize/System Drive Size (MB)	rWin32_Processor
		wFreeSpace/System Drive Free (MB)	rWin32_LogicalMemoryConf
		wSystemType/System	
		wManufacturer,	
		wCurrentClockSpeed/Processor	
		wTotalPhysicalMemory/Physical Memory (MB)	
		wVersion/Bios	
Status - Application Events	Application Events	device_id/Subscriber	AppEvent

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		service_id/Service	
		ctime/Created	
		mtime/Modified	
		app_name/Application Name	
		event/Event	
		status/Status	
		del_time/Date Deleted	
		ver_time/Date Verified	
		inst_time/Date Installed	
		fix_time/Date Fixed	
Status - Connect	Connections	mtime/Modified	DeviceStatus
		device_id/Subscriber	
		duration/Duration	
		mrc/Return Code	
		reason/Reason	
		rsrc_transfer/File Tx (#)	
		rsrc_transfer_size/Files Tx (Sz)	
Status - Notify	Notify Queue	mtime/Modified	DeviceNotify
		device_id/Subscriber	
		nfy_status/Status	
		nfy_reason/Reason	
		nfy_type/CommsType	
		nfy_attempts/Attempts (#)	
Errors - Connect	Connect Errors	mtime/Modified	DeviceErrors
		device_id/Subscriber	
		type/Type	
		code/Code	
		reason/Reason	
Errors - Notify	Notify Errors	mtime/Modified	DeviceNotify
		device_id/Subscriber	
		nfy_attempts/Attempts (#)	
		nfy_status/Status	
		nfy_reason/Reason	
		nfy_type/Comms Type	

Table 29   Inventory Reporting – Multicast Detail Reporting

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
Status – Multicast Server Statistics	Multicast Server Statistics	mtime	rNVD_MulticastStatistics
		userid	
		wDuration/Transmit Duration	
		wNamespace	
		wNbytesRej	
		wNbytesReq	
		wNbytesXmt/Bytes Transmitted	
		wNclients/Agents Connected	
		wNdevices	
		wNfilesRej/Files Rejected	
		wNfilesReq/Files Requested	
		wNfilesXmt/Files Transmitted	
		wServiceID/Service	
		wSourceID/Multicast Session	
		wSourceType	
		wStartTime/Transmit Start	
Status – Agent Download Statistics	Agent Downlaod Statistics	mtime	RNVD_DownloadStatistics
		userid/Subscriber	
		wDuration/Transmit Duration (sec)	
		wNamespace	
		wNbytesRcv/Bytres Received	
		wNbytesRej	
		wNbytesReq	
		wNfilesRej/FilesRejected	
		wNfilesRcv/Files Received	
		wNfilesReq/Files Requested	
		wNpktsDrp	
		wNpktsRcv	
		wServiceID/Service	
		wSourceID	
		wSourceType/Source Type	
		wStartTime/Transmit Start	

B 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 30 below will help you identify naming changes that have been applied to the Radia brand.

Table 30   Product Name and Term Changes

New Name/Term	Old Name/Term
CM Agent Installation Wizard	Radia Client Installation Wizard
CM agents	Radia clients
HP Configuration Management Administrator	Radia Administrator Workstation
HP Configuration Management	Radia
HP Configuration Management Admin Agent Explorer	Radia Client Explorer
HP Configuration Management Admin CSDB Editor	Radia System Explorer
HP Configuration Management Admin Packager	Radia Packager
HP Configuration Management Admin Publisher	Radia Publisher
HP Configuration Management Admin Screen Painter	Radia Screen Painter
HP Configuration Management Application Manager	Radia Application Manager,
HP Configuration Management Application Self-service Manager	Radia Software Manager
HP Configuration Management Application Usage Manager	Radia Usage Manager
HP Configuration Management Configuration Server	Radia Configuration Server

New Name/Term	Old Name/Term
HP Configuration Management Configuration Server Database	Configuration Server Database, Radia Database
HP Configuration Management Inventory Manager	Radia Inventory Manager
HP Configuration Management Messaging Server	Radia Messaging Server
HP Configuration Management OS Manager	Radia OS Manager
HP Configuration Management Patch Manager	Radia Patch Manager
HP Configuration Management Reporting Server	Radia Reporting Server
HP Configuration Management Solutions for Servers	Server Management

Index

repairing
    CM agent, 63
    silent installation, 57

Common Information Model, 20
    definition, 18

connect.bak, 48

copyright notices, 2

CREATE method, 140

CUSTOM, 48, 51

customer support, 7

D

DATE attribute, 94

db_import directory, 81

DEFERRED, 138

Delete Discovered Application Component, 109

DELETE method, 140

Desktop class, 74

DESKTOP class, 74

Differenced Audit on Install and Update, 122

Differenced Audit on Install, Verify, and Update, 122

DIRPATH attribute, 94

DISCOVER attribute, 53

Disk Cost, 60

Distributed Management Task Force, 21

document changes, 4, 5

documentation updates, 4

DOMAIN attribute, 140

DRIVE attribute, 94

E

enterprisemanagement argument, 47

EXECUTE.REXX, 146

F

file auditing, 88

File class, 74

FILE class, 74

File Scanner class, 74

File Scanner Filters class, 74

FILEAUDT object, 92, 93, 115, 117
    ACCESSDT, 93
    ACCESSTM, 93
    ATRIBUTE, 93
    DATE, 94
    DIRPATH, 94
    DRIVE, 94
    FULLPATH, 94
    PAHTCRC, 94
    RESOLVE, 94
    SIZE, 94
    STATUS, 94
    TIME, 94
    XTENSION, 94

FILENEW object, 116

FILEPREV object, 115

FILESCAN class, 74

FILTER class, 74

filtering patch reports, 159

Full Audit on Install and Update, 122

Full Audit on Install, Verify and Update, 122

FULLPATH attribute, 94

G

GATEWAY attribute, 102

GROUP attribute, 90

H

HALCOMP attribute, 102

HALDATE attribute, 103

HALFNAME attribute, 103

HALFVER attribute, 103

HALINAME attribute, 103

HALLANG attribute, 103