

HP Client Automation Inventory Manager

for the Windows® operating system

Software Version: 8.10

Reference Guide

Document Release Date: February 2012
Software Release Date: February 2012



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 Notice

© Copyright 1993-2011 Hewlett-Packard Development Company, L.P.

Trademark Notices

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

UNIX® is a registered trademark of The Open Group.

Acknowledgements

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Documentation Updates

The title page of this document 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 of a document, go to:

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

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

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

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

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

Support

Visit the HP Software Support Online 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 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 web 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 register for an HP Passport user ID, go to:

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

To find more information about access levels, go to:

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

Contents

1	Introduction	7
	Overview	7
	Inventory Manager Terminology	7
	Client Automation Prerequisites	8
	Necessary Skills	8
	Inventory Manager Technology	9
	Common Information Model (CIM)	9
	Web-Based Enterprise Management (WBEM)	9
	Windows Management Instrumentation (WMI)	10
	Client Automation and WBEM	10
2	The AUDIT Domain	11
	The AUDIT Domain in the CSDB	11
	AUDIT Domain Defined	12
	RIMOPTS Class	14
	REGISTRY Class	15
	Implementing Registry Scans	17
	Inventory Database Tables	17
3	Software and Hardware Auditing	19
	Auditing Types	19
	File Auditing	19
	WBEM Auditing	24
	WBEM Object Processing	27
	Disabling Remnant Configuration Server Instances for WBEM Object Processing	27
	Hardware Auditing	27
4	Successful Auditing	33
	Sample Auditing	33
	Configuring a Sample Audit	36
	What Happened during the Scan?	40
5	Creating Audit Packages	43
	Audit Packages or PACKAGE Class	43
	Using the CSDB Editor Create/Maintain Audit Services	45
6	Configuring Timers for Audit Collection	55
	The Scheduling (TIMER) Class	55
	Creating a Timer Instance	60
	Specifying Timer Settings	62

Specifying ZSCHDEF	62
Specifying ZSCHTYPE	63
Specifying ZSCHFREQ	63
Specifying ZRSCCMDL	63
Specifying ZNOPING, PINGDLAY, and PINGCNT	64
Connecting the Timer to a Service	64
Audit Execution Configuration	65
7 Viewing Inventory Reports	71
Reporting Views for Inventory Reports	71
Windows Vista Readiness Reports	73
Filtering Inventory Reports with Reporting Server	74
A Detail and Summary Reporting Tables	77

1 Introduction

Overview

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

Systems administrators use the HP Client Automation Administrator Configuration Server Database Editor (Admin CSDB Editor) to specify what inventory management data is to be collected. An Inventory connect (DNAME=AUDIT) is then run on the target computer to retrieve the desired information and send it up to the HPCA server for later reporting.

For more information on reporting, refer to the *HP Client Automation Core and Satellite Enterprise Edition User Guide*.

Inventory Manager Terminology

Agent Computer

Agent computer is the computer on the end user's desktop that has the Client Automation agent software installed on it.

CIM (Common Information Model)

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

Clean Machine

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

Client Automation agent

Client Automation agent is the Client Automation software component that is installed on the end user's desktop computer.

Messaging Server

The Messaging Server is the Client Automation infrastructure component that provides a common routing and inter-server data delivery service, especially for report-bound data. When servicing a Configuration Server, the Messaging Server handles the delivery of

Inventory, Patch, and other data collected from Client Automation agents to the appropriate external location. Data Delivery Agents are used to post data directly to an SQL-compliant database using ODBC.

Reporting Server

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

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.

Client Automation Prerequisites

The Inventory Manager requires the following Client Automation components:

- HPCA Server
- Client Automation agent
 - Application Manager with Inventory Manager feature
 - Application Self-Service Manager (optional)
- HPCA Administrator CSDB Editor. This is installed as part of the HPCA Administrator. For more information on HPCA Admin CSDB Editor, refer to the *HP Client Automation Administrator Installation and User Guide*.

Necessary Skills

Client Automation Modules

This document assumes that the reader is familiar with the CSDB, with administering it using the CSDB Editor and the Enterprise Manager. See the *HP Client Automation Administrator Installation and User Guide* and the *HP Client Automation Core and Satellite Enterprise Edition User Guide* for more information.

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> .

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.

Inventory Manager Technology

While an administrator with little web-based knowledge can use the Inventory Manager with success, it is important to understand some of the technology behind the product. The information that is provided below is intended to give you a preliminary understanding of the technology behind the Inventory Manager agent. As indicated in [Necessary Skills](#) on page 8, 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.



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

Client Automation and WBEM

The Inventory Management agent queries the WBEM namespace (that is, the WBEM database) and sends the results back to the Configuration Server. All information collected by WBEM is available to the Inventory Manager agent. The collected information is then stored in the ODBC inventory database.

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

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



Inventory Manager for Windows 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 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.

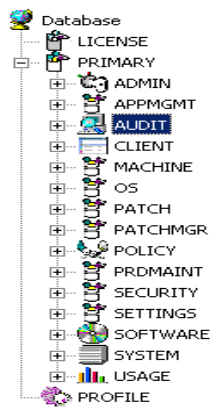
2 The AUDIT Domain

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

The AUDIT Domain in the CSDB

The AUDIT Domain is located in the PRIMARY File of the 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 figure below shows its tree structure in the CSDB Editor.

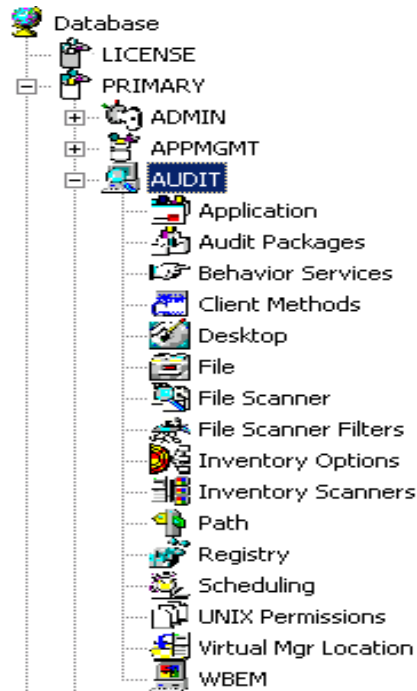


Table below describes the classes in the AUDIT Domain.

Table 1 AUDIT Domain

Class	Description
Application (ZSERVICE)	These are sample services distributed with the Inventory Manager. The AUDIT.ZSERVICE instance is connected to a policy instance. A policy instance can be an instance of the Users, Departments, or Workgroups class. It can also be a customer-defined class within the POLICY Domain. Each of the sample ZSERVICE classes is connected to the PACKAGE instances.
Audit Packages (PACKAGE)	Defines what information to collect, and then what actions to take. These packages would contain various audit components. A good example is an audit of running services on a desktop. The AUDIT.ZSERVICE instance must contain a connection to an AUDIT.PACKAGE instance.
Behavior Services (BEHAVIOR)	Defines instances that enable the execution of auditing on the agent. Normally, there is no need to add or modify instances in this class.
Client Methods (CMETHOD)	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.

Table 1 AUDIT Domain

Class	Description
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 14.
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.
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 Configuration Server during a FILE audit.
WBEM (WBEM)	This class contains instances that define Inventory Manager scans of WMI classes. These can include any class in the WMI database such as Win32_Services. This example would provide information on Windows NT or Windows 2000 services.

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 below describes these attributes.

Table 2 RIMOPTS Class

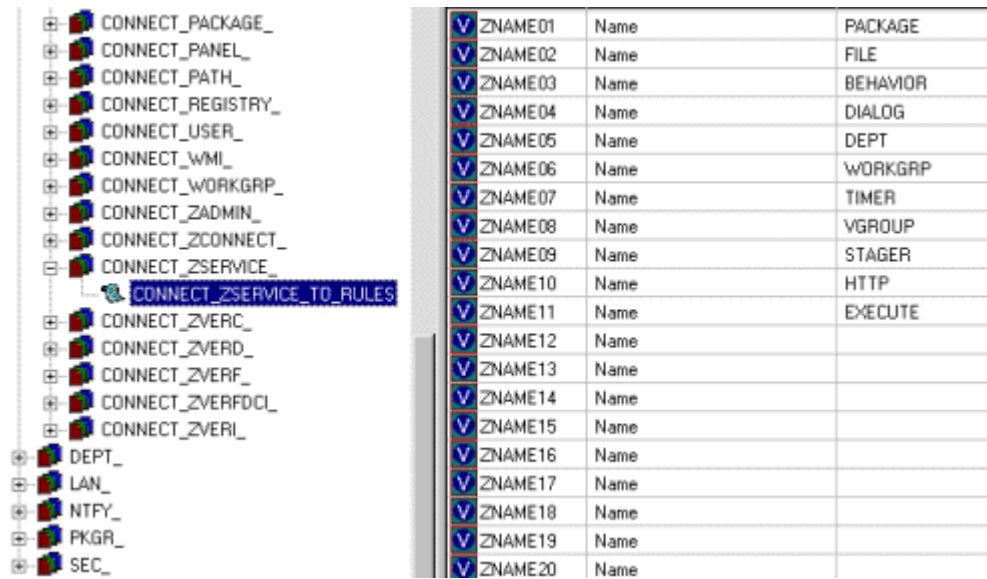
Attribute	Usage
COLLECT	Audit Collection Type by selecting Diff or Full . <ul style="list-style-type: none">Select Diff to report the difference between the previous information collected for the service and the information collected during the current agent audit. This is the default setting. <p>Note: The first or initial scan of the DIFF setting will be a FULL scan as defined below. All subsequent scans will then be differenced unless the administrator changes the setting to FULL.</p> <ul style="list-style-type: none">Select Full to report the information collected for the service during the current agent connect process without differencing against the previous collection for that service.
RUNEXEC	This string indicates what actions the Inventory Manager will take upon connection: <ul style="list-style-type: none">Select I to invoke collection of information when the service is installedSelect 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 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.

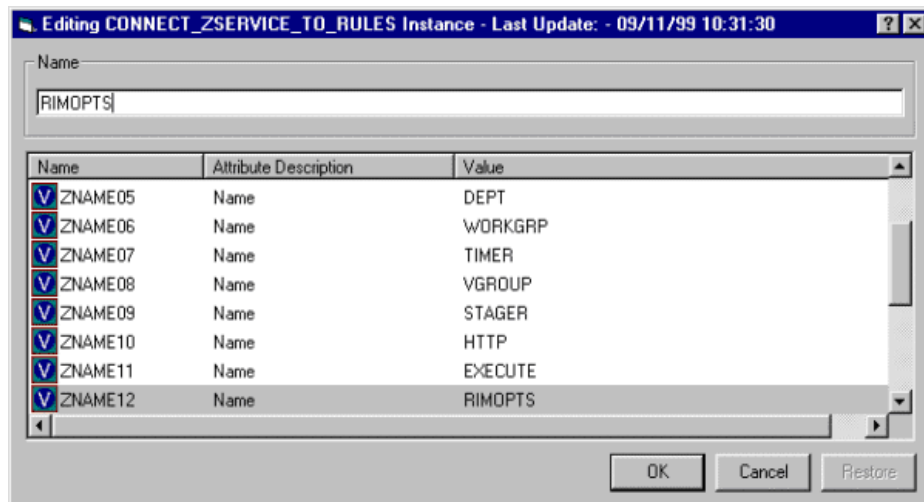
Prior to beginning any tasks using the Inventory Manager, you must enable the drag-and-drop feature for the newly created RIMOPTS Class instances. For additional information about editing instances, see the *HP Client Automation Administrator Installation and User Guide*.

To Enable Drag-and-Drop Connections for RIMOPTS Class Instances

- 1 Open the Admin CSDB Editor and go to **PRIMARY** → **ADMIN** → **Name Lists (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 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 3 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.
NAMESPACE	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.
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 <i>n</i> <ul style="list-style-type: none"> • Set to 0 to only scan current subkey. • Set to -1 to scan all subkey levels. • Set from 1 through <i>n</i> 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 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 in a few steps:
 - a Use the **Available Components** drop-down list to select **Registry**.
 - b In the **New Component Name** 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.











































































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 Messaging Server, which posts it to the ODBC database for inventory.

- 6 View the Registry Scan report from the Reporting Server.

Inventory Database Tables

The inventory reporting database includes the tables shown in the following figure, among others.

Figure 1 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-Wbem sources.

The recommended product for viewing Inventory is the Reporting Server. See the *HP Client Automation Reporting Server Reference Guide* for more information.

3 Software and Hardware Auditing

This guide is provided to assist you with installing and implementing the 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 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 40 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 4 AUDIT.FILE Class Instances

Attribute	Description and Examples
SCANFOR	<p>Indicate a fully qualified path and file name to search for. Wildcards are permitted.</p> <p><SystemDrive:>\WinNt**.dll</p>
ACTION	<p>The RIMDIFF method performs actions on the files discovered on the user's computer during the agent connect.</p> <ul style="list-style-type: none"> • Y configures RIMDIFF to perform the action. • N configures RIMDIFF to not perform the action. <p>The first four flags determine <i>when</i> to report that the files were found: Report on: Initial, New, Changed, Deleted</p> <ul style="list-style-type: none"> • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan. • Deleted means that the file was found during the previous scan. The file is not present for the current scan. <p>The last three flags control the <i>actions to take</i> on the files detected during the current scan. Action to take on discovery: Send, Delete, Custom</p> <ul style="list-style-type: none"> • Send means to send the files to the Configuration Server and store them in the location indicated by the ZRSCVLOC attribute (see ZRSCVLOC on page 21). • Delete means to delete the files from the user's computer. • Custom means to execute the method indicated in the CUSTOM attribute. <p>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.</p>
OUTPUT	Output object name.
TYPE	Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM. File.
GROUP	Optional way to identify a set of scan results. This maybe useful for querying and reporting on the audited files from the database where audit results can be stored. Games, MPEGs.

Table 4 AUDIT.FILE Class Instances

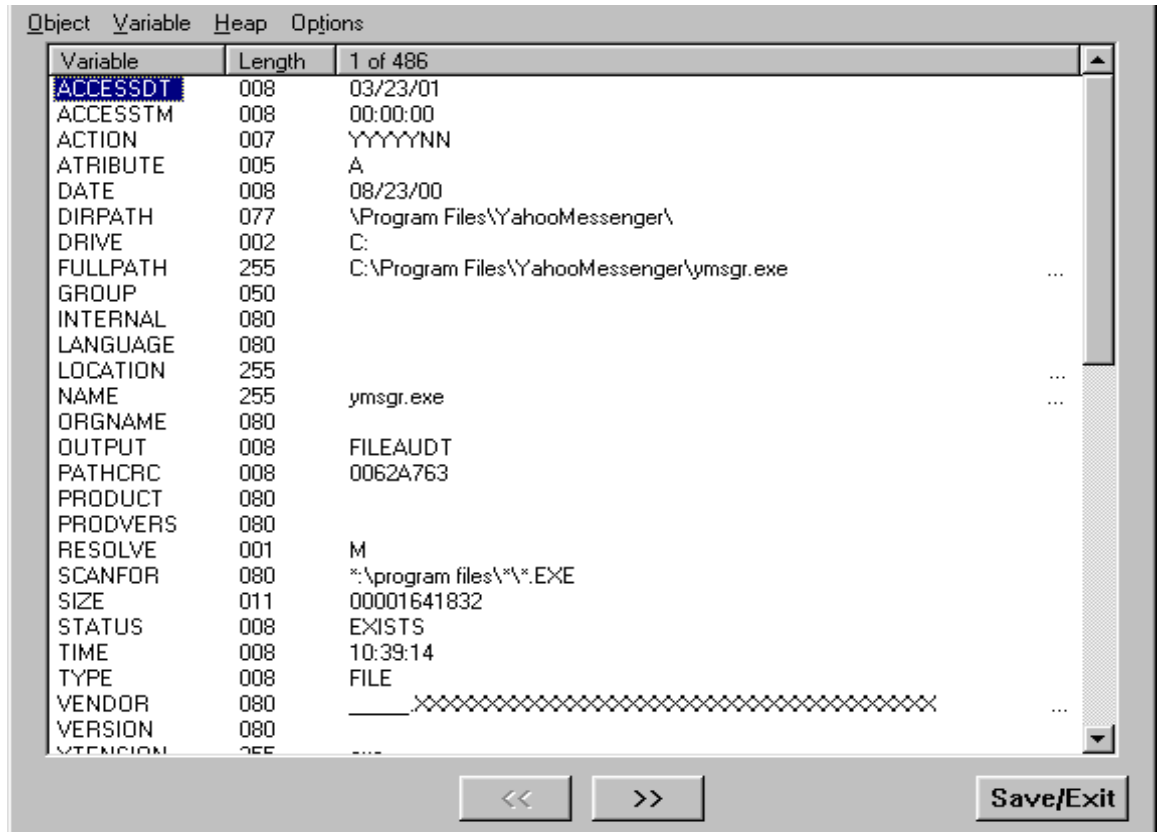
Attribute	Description and Examples
ZVERINFO	<p>Collect extended information.</p> <ul style="list-style-type: none"> • Set the value to 1 to collect additional information for a file. • Set the value to 0 to not collect additional information. <p>In order for this data to be collected, the associated attribute must exist in the AUDIT.FILE class template.</p> <p>You can limit the scan to only those files that have some particular values in their extended information. You do so by supplying a value (either 1 or 0) for any of the associated attributes in an AUDIT.FILE instance. This causes the scan to be filtered. Only those files whose extended information data element contains the value you specify in its associated attribute will be scanned.</p> <p>Extended file information consists of one or more of the following data elements. The associated attribute name for the data element is in parentheses:</p> <ul style="list-style-type: none"> • (VENDOR) – The seller of the file/product • (PRODUCT) – The name of the item for which the file is a part. • (PROVERS) – The version of the product which the file is a part. • (ORGNAME) – The name of the organization. • (INTERNAL) – The internal data element encoded in the file. • (VERSION) – The version of the file. • (LANGUAGE) – The language of the file.
ZRSCSTYP	Server file type. This can be either Binary or Text. The administrator does not set this.
ZRSCMFIL	Manager directory location.
ZRSCVLOC	<p>The location on the Configuration Server where the files are stored because of the Send Action (see ACTION on page 20). This variable needs to be configured when sending a file back to the Configuration Server. The variable should contain the name of the MGRVLOC instance that will be used to resolve the location to store the uploaded file.</p> <p><SystemDrive:>\Data\&(ZOBJPID) \&(name)</p>
ZRSCMEM	PDS member name. This field is optional.
PRODUCT	<p>The product name.</p> <p>See ZVERINFO on page 21 for more detail.</p>
PROVERS	<p>The product version.</p> <p>See ZVERINFO on page 21 for more detail.</p>
ORGNAME	<p>The organization name.</p> <p>See ZVERINFO on page 21 for more detail.</p>
INTERNAL	<p>The internal data element encoded in the file.</p> <p>See ZVERINFO on page 21 for more detail.</p>
VERSION	<p>The version of the file.</p> <p>See ZVERINFO on page 21 for more detail.</p>

Table 4 AUDIT.FILE Class Instances

Attribute	Description and Examples
LANGUAGE	The language of the file. See ZVERINFO on page 21 for more detail.
VENDOR	The product vendor. See ZVERINFO on page 21 for more detail.
ZRSCCRC	Resource CRC.
ZCRCINFO	Collect file CRC. [Y/N] Default is N . <ul style="list-style-type: none"> • 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 . <ul style="list-style-type: none"> • 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, that is, Publisher.
ZINIT	Not applicable at this time.
NAME	Not applicable at this time.
LOCATION	Not applicable at this time.

Use the Agent Explorer to view the FILEAUDT object results as shown in following figure.

Figure 2 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 5 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.
ATTRIBUTE	A string listing the attributes of the file: R = Read only A = Archive S = System H = Hidden C = Compressed
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.

Table 5 FILEAUDT Object

Attribute	Description
PATHCRC	A unique number that indicates the CRC path used for differencing.
RESOLVE	The value of M indicates that the 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: <ul style="list-style-type: none"> • Exists means that this is the first time scanning for this file and it was found. • New means that this file was added to the file system of the agent computer since the last scan was performed. • Update means that this file exists in the new and previous scans. There have been changes to the date, time, size, and/or version. • Deleted means that this file was present in the previous scan but is missing in the new scan. • Not found means that 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

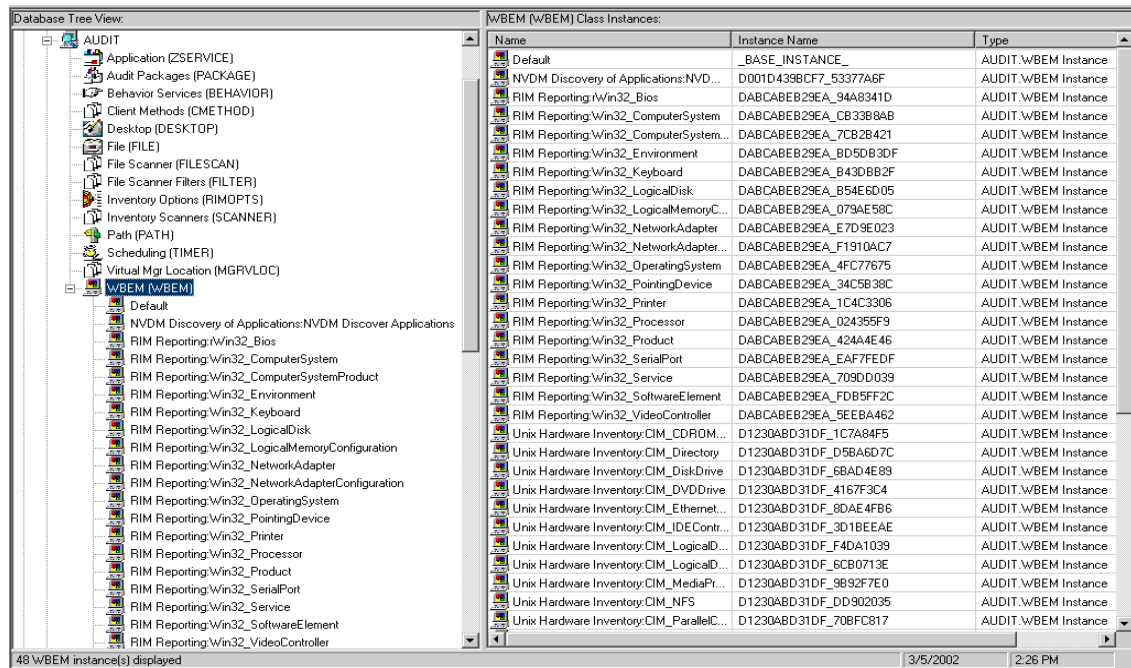
Use the RIMWBEM method to query the WBEM namespaces to retrieve information about how a system's hardware and software is used. The RIMWBEM method constructs a query from the information contained in an instance of the AUDIT.WBEM class. WBEM has a query engine that processes the query statement and returns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



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 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 3 AUDIT.WBEM class instances



Following table describes the attributes of the AUDIT.WBEM instance.

Table 6 AUDIT.WBEM Instance

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

Table 6 AUDIT.WBEM Instance

Attribute Name	Description
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 Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.
NAME	Friendly name for this instance. This name will appear in the 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 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 Configuration Server.

In addition to the attributes described in [Table 6](#) on page 25, the WBEM object also contains the following:

Table 7 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 Configuration Server.
ZOBJRSIZ	The resource size maintained by the Configuration Server.
ZOBJTIME	The latest time under the current node.
ZRSCSRC	The name of the program promoted the resource.

WBEM Object Processing

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

- 1 At the end of the agent connect, the ZTASKEND REXX method on the Configuration Server is called and creates commands to invoke the QMSG executable.
- 2 QMSG.EXE places the WBEMAUDT objects into the Configuration Server `\data\wbem` directory, or message queue.
- 3 The Messaging Server includes a WBEM Data Delivery Agent (WBEM.DDA) that monitors this `\data\wbem` message queue and processes the WBEM objects.
- 4 The WBEM.DDA is usually configured to post the WBEM objects directly to an ODBC-compliant Inventory Manager database, or, it may be configured to first forward the WBEM objects to another Messaging Server located closer to the database. In the later case, the receiving Messaging Server posts the WBEM data to the Inventory ODBC-compliant database.
- 5 After it is posted to the Inventory Manager database, the new WBEM information is immediately available for query and reporting purposes through the Reporting Server.

For more information, see the *HP Client Automation Enterprise Messaging Server Reference Guide*.

Disabling Remnant Configuration Server Instances for WBEM Object Processing

Inventory Manager no longer supports processing WBEM objects using these instances in the CSDB:

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

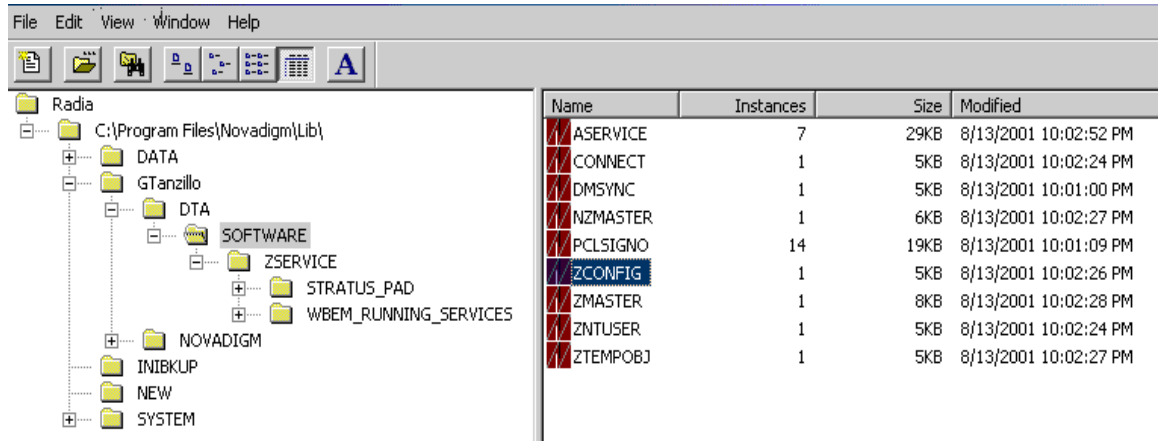
If these remnant instances exist or were imported into your CSDB, you must disable any configurations within them in order to ensure successful WBEM object processing.

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

Hardware Auditing

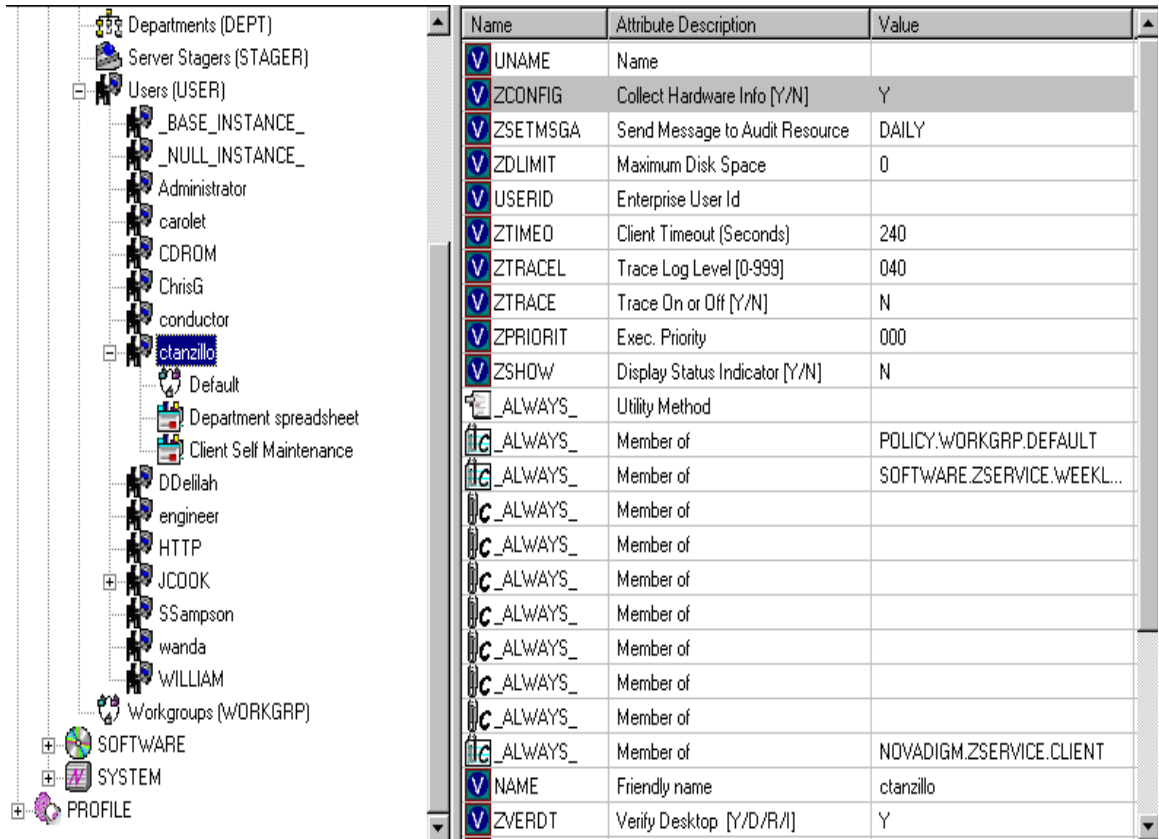
Each time a Client Automation agent connects to the Configuration Server, information about the agent's hardware configuration is stored in the ZCONFIG object. The ZCONFIG object is calculated and stored in the application service directory of the Client Automation agent's object directory tree as follows:

Figure 4 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 5 POLICY.USER Class – ZCONFIG Attribute



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

Figure 6 ZCONFIG Object

Variable	Length	Options
GATEWAY03	013	208.244.231.1
IPADDR01	007	0.0.0.0
IPADDR02	007	0.0.0.0
IPADDR03	015	208.244.231.104
LADAPT01	012	444553540000
LADAPT02	012	444553540001
LADAPT03	012	0050da644154
REBOOTD	008	20010608
REBOOTT	008	11:15:40
SUBNET01	007	0.0.0.0
SUBNET02	007	0.0.0.0
SUBNET03	013	255.255.255.0
ZGATEWAY	011	&[GATEWAY03]
ZHDWBIOS	037	04/22/99 PhoenixBIOS 4.0 Release 6.0
ZHDWCDDR	002	E:
ZHDWCOMP	009	ctanzillo
ZHDWCPU	007	Pentium
ZHDWCPU\$	006	450MHz
ZHDWD00	002	C:
ZHDWD00C	005	Fixed
ZHDWD00F	014	10,109,403,136
ZHDWD00S	005	FAT 32
ZHDWD00T	014	13,689,888,768
ZHDWD01	002	E:
ZHDWD01C	005	CDROM
ZHDWD02	002	F:
ZHDWD02C	006	Remote
ZHDWD02F	011	541,196,288
ZHDWD02S	007	NWCOMPA
ZHDWD02T	013	1,048,576,000
ZHDWD03	002	G:
ZHDWD03C	006	Remote
ZHDWD03F	014	51,560,579,072
ZHDWD03S	004	NTFS

The ZCONFIG object stores hardware information discovered by the Client Automation 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 8 ZCONFIG Object

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

Table 8 ZCONFIG Object

Attribute	Description
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 reboot date
REBOOTT	Last reboot time
SUBNET##	Subnet mask.
ZGATEWAY	Looks at GATEWAY attribute
ZHDWBIOS	BIOS type.
ZHDWCDDR	Client Automation 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
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

Table 8 ZCONFIG Object

Attribute	Description
ZHDWPSER	Number of serial ports
ZHDWVIDEO	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 Client Automation agent connect for this service
ZOBJNAME	HARDWARE_SCAN (hard coded)
ZOBJTIME	The time of the agent connect
ZSUBNET	The subnet mask
ZUSERID	The name of the user who connected

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

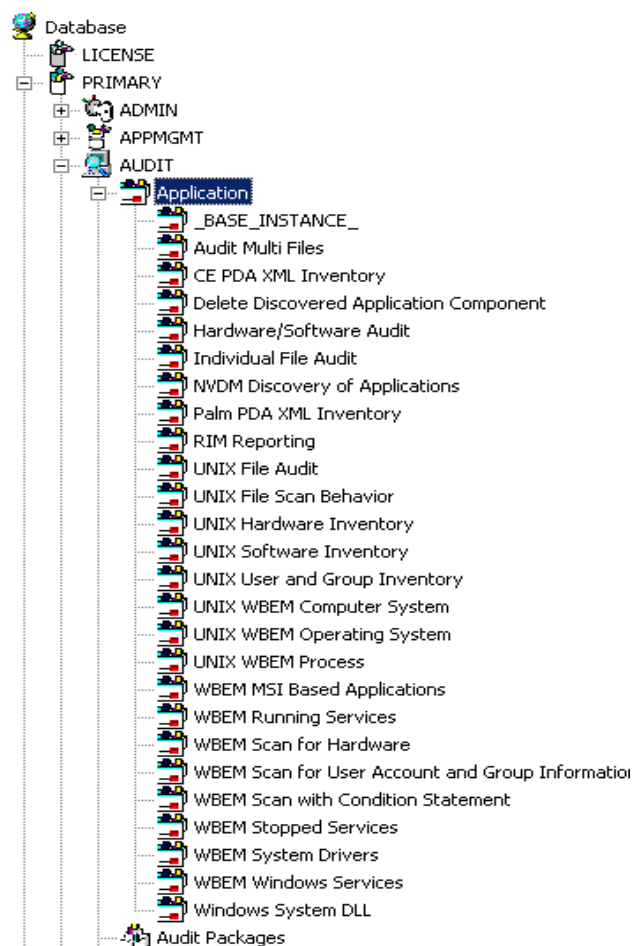
4 Successful Auditing

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

Sample Auditing

To illustrate the concepts of inventory information collection, the Inventory Manager installation contains a set of representative audit service examples. These samples are located in the PRIMARY.AUDIT.Application (ZSERVICE) class as follows:

Figure 7 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 Inventory Manager upgrade.

The sample audit services are described in the following table:

Table 9 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 Client Automation 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.
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.

Table 9 Sample of Auditing Services

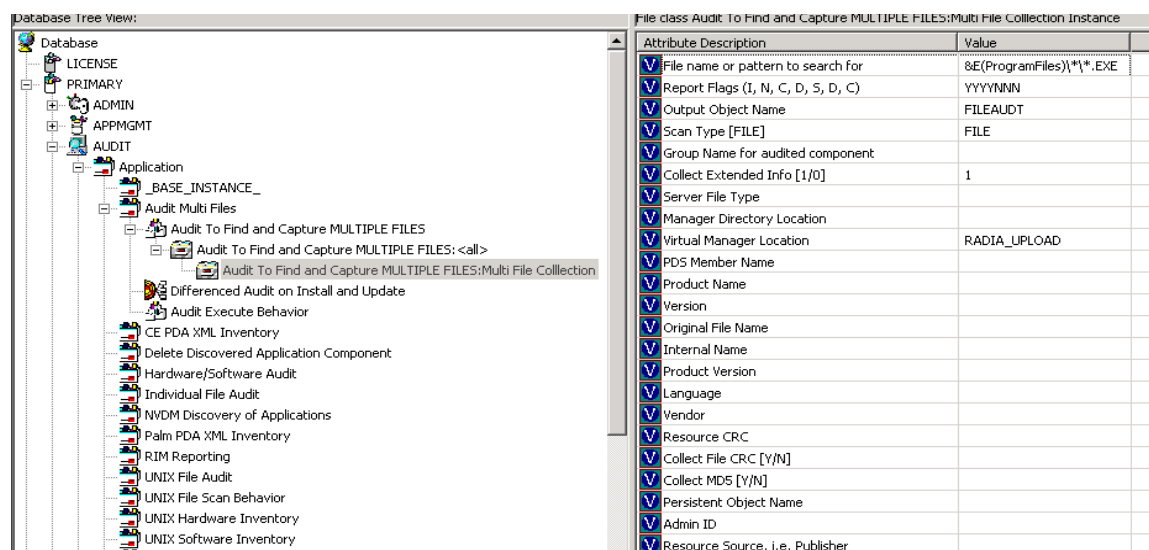
Service	Connected to Audit Package (PACKAGE)	Description
Unix File Scan Audit	UNIX File Scan Audit	This service performs a NVDM scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes on UNIX platforms.
Unix Hardware Inventory	Unix Hardware Inventory	This service scans 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.
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. See the *HP Client Automation Administrator Installation and 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 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 Configuration Server for storage.

Figure 8 SCANFOR Attribute of the Audit Multi Files Instance

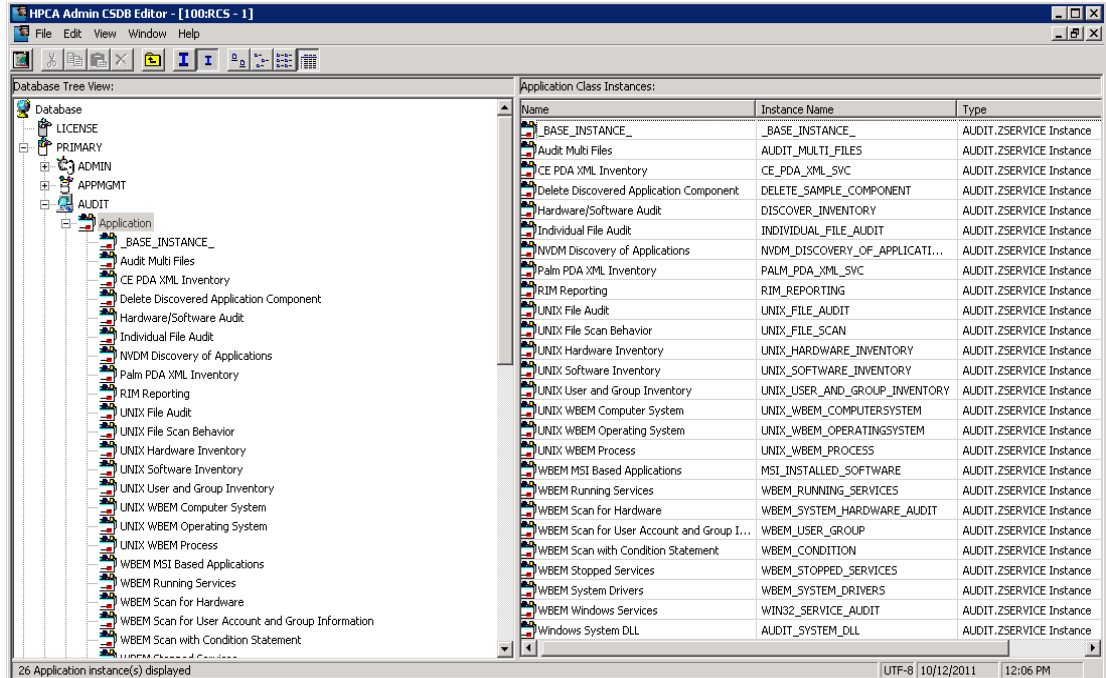


Attribute Description	Value
File name or pattern to search for	&E(ProgramFiles)**.EXE
Report Flags (I, N, C, D, S, D, C)	YYYYNNN
Output Object Name	FILEAUDT
Scan Type [FILE]	FILE
Group Name for audited component	
Collect Extended Info [1/0]	1
Server File Type	
Manager Directory Location	
Virtual Manager Location	RADIA_UPLOAD
PDS Member Name	
Product Name	
Version	
Original File Name	
Internal Name	
Product Version	
Language	
Vendor	
Resource CRC	
Collect File CRC [Y/N]	
Collect MDS [Y/N]	
Persistent Object Name	
Admin ID	
Resource Source, i.e. Publisher	

To Configure a Sample Audit Package

- 1 If you have not already done so, start the CSDB Editor.
- 2 Navigate to and expand the PRIMARY.AUDIT Domain.

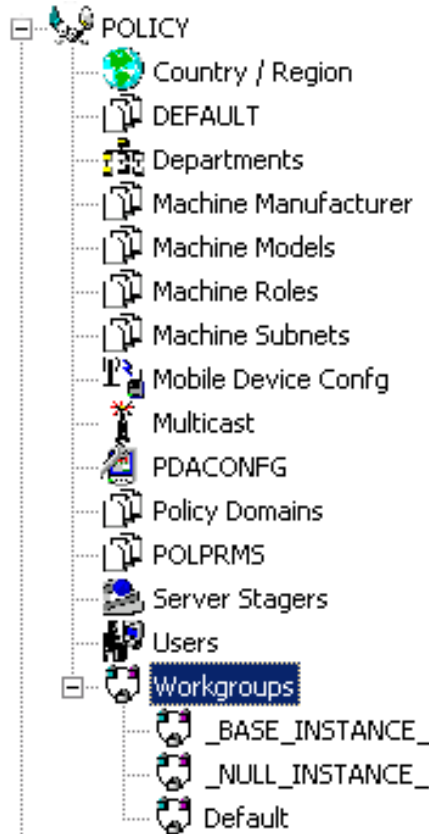
- 3 Double-click **Application (ZSERVICE)** to expand the class.



- 4 Scroll to and expand the **POLICY** Domain.

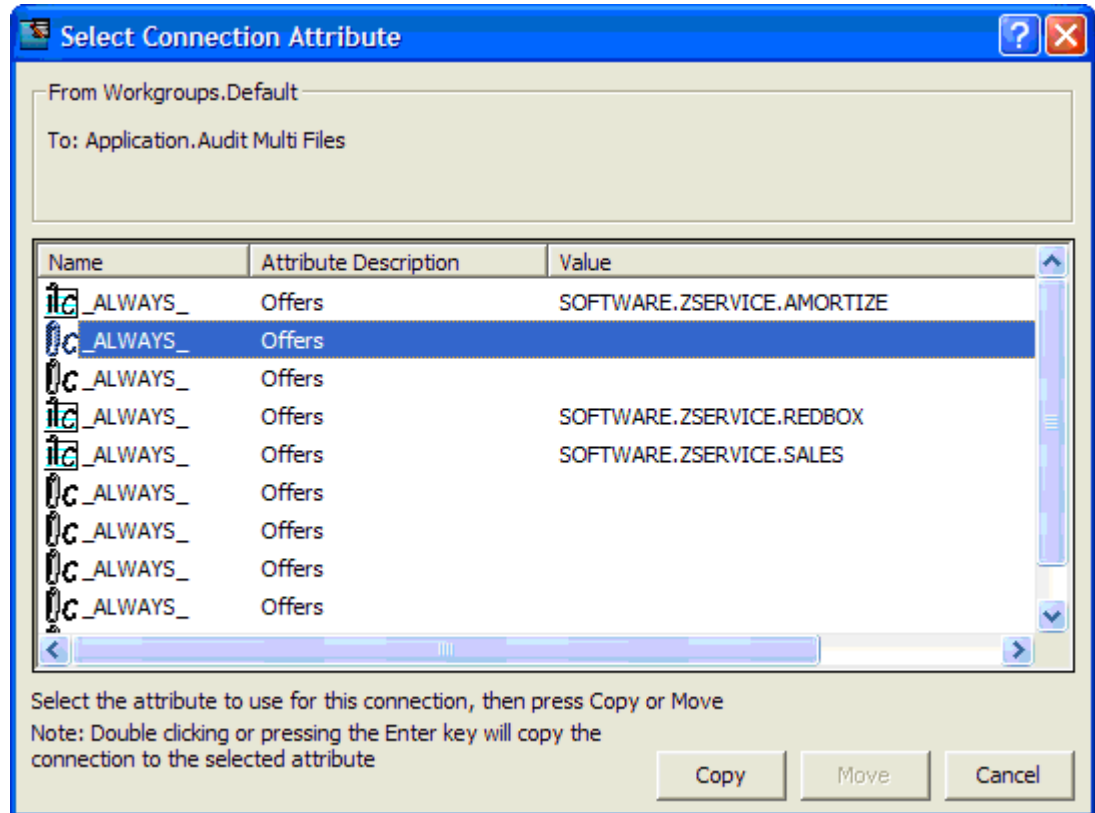
For our example, we would like all users that are members of the Workgroup class to select this audit package from their Application Self-Service Manager.

- 5 Expand the **POLICY.WORKGROUPS** class.



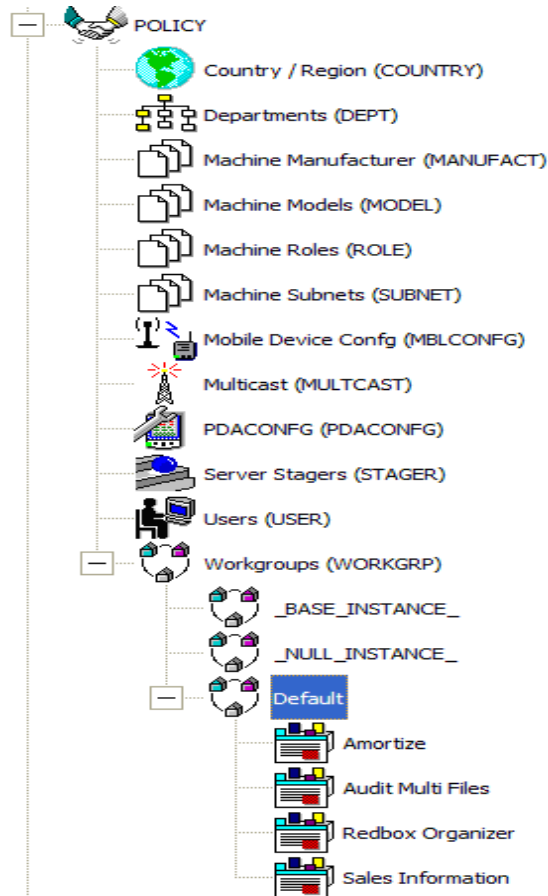
- 6 Select the **Audit Multi Files** package from the ZSERVICE class and drag it to the POLICY.WORKGROUPS class and drop it on the **Default** instance.

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 Inventory Manager agent computer when a user connects to the Configuration Server as follows:

- Through an Application Self-Service Manager agent connect, when the user launches that program.

or

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

The following figure shows the available Audit Multi Files package that an Application Self-Service Manager user would see when connecting to the Configuration Server:

Figure 9 Application Self-Service Manager shows Audit Multi Files

Name	Status	Compressed Size	Description	Mandatory	
Amortize	Available	n/a		0	
Audit Multi Files Version 1.0					Size 6.67 KB Compressed Size 2.34 KB
Available					i
Drag & View	Available	2.51 MB		0	
G5-CALC	Available	n/a		0	
Redbox Organizer	Available	n/a		0	
Sales Information	Available	n/a		0	
StratusPad	Available	n/a		0	

When the subscriber selects and installs the Audit Multi Files package from the 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 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 Agent Explorer to locate the ZSERVICE for the Audit Multi Files package in the LIB directory.

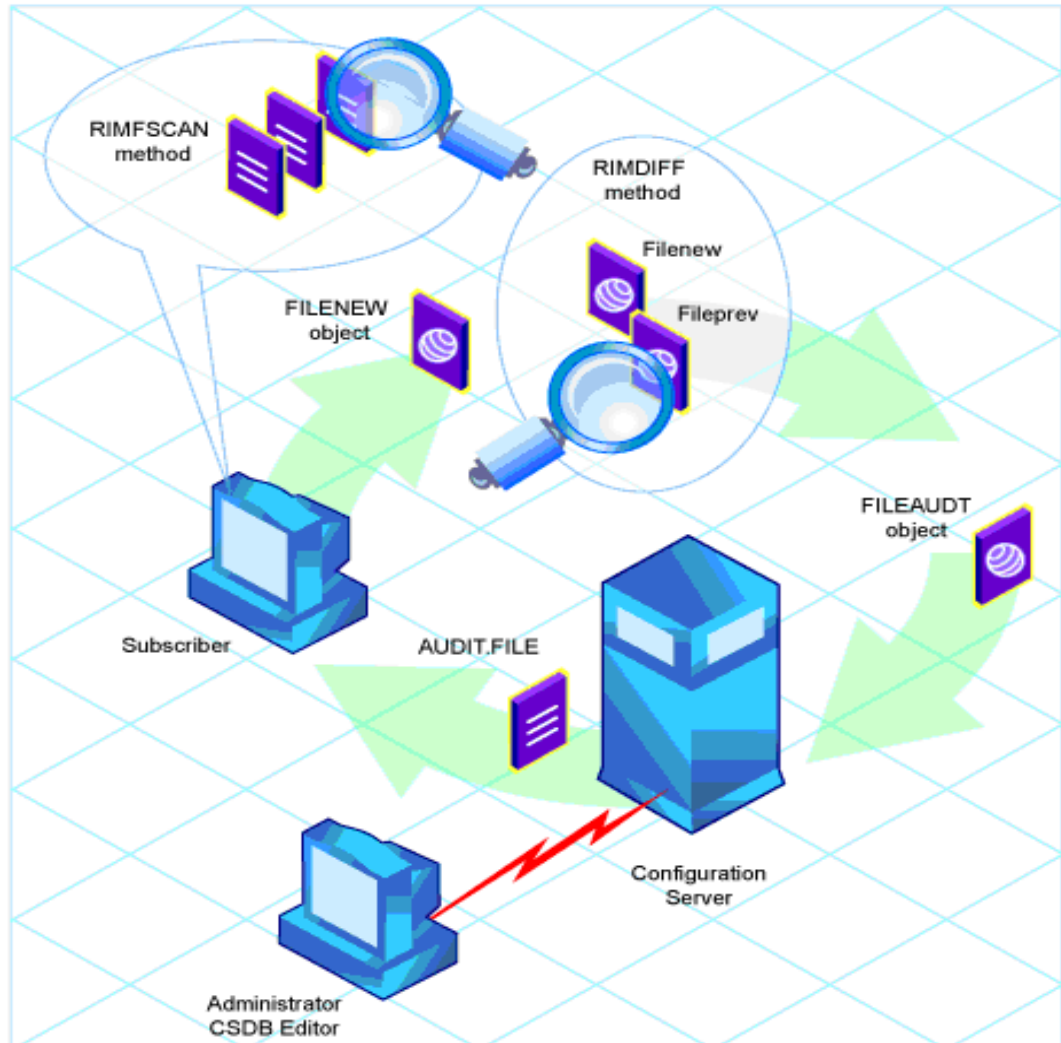
Figure 10 AUDIT_MULTI_FILES in the LIB Directory

Name	Instances	Size	Modified
APPINFO	1	6KB	8/13/2001 10:10:19 PM
CONNECT	1	5KB	8/13/2001 10:10:19 PM
DMSYNC	1	5KB	8/13/2001 10:10:19 PM
FILEAUDT	370	1MB	8/13/2001 10:11:05 PM
FILEPREV	370	1MB	8/13/2001 10:11:05 PM
PCLSIGNO	14	19KB	8/13/2001 10:10:19 PM
ZCONFIG	1	5KB	8/13/2001 10:10:20 PM
ZMASTER	1	6KB	8/13/2001 10:13:07 PM
ZNTUSER	1	5KB	8/13/2001 10:10:19 PM

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 11 Auditing with the RIMFSCAN and RIMDIFF Methods



- The `RIMFSCAN` method scans the Client Automation 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 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 Client Automation agent's computer.

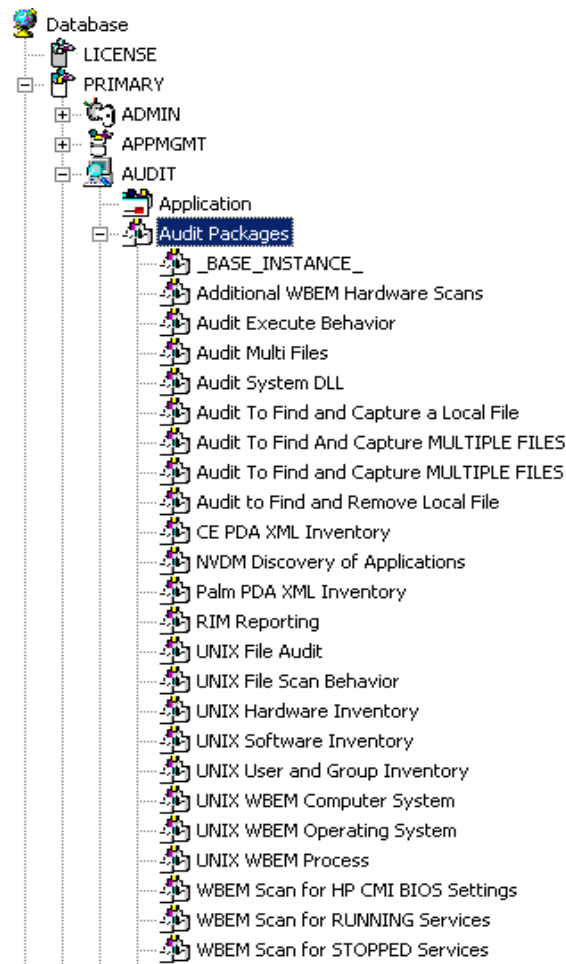
5 Creating Audit Packages

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.

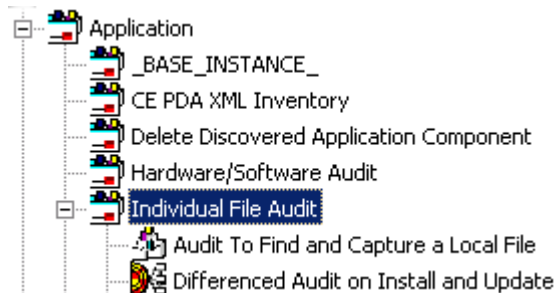
By expanding the Audit Packages (PACKAGE) class, you will see the audit package instances.

Figure 12 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 the **Individual File Audit** instance.

Figure 13 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 10 Inventory Options (RIMOPTS) Class

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

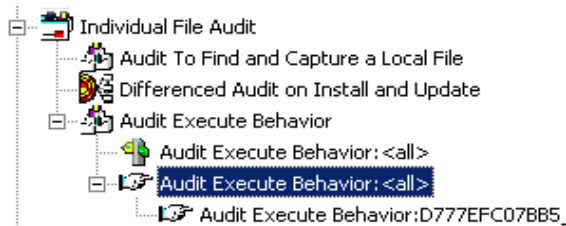
Table 10 Inventory Options (RIMOPTS) Class

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

for additional information about RIMOPTS attributes.

Finally, a connection to an auditing behavior is needed.

Figure 14 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 *HP Client Automation 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

We will use the CSDB Editor to 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 55 for information concerning timers.)

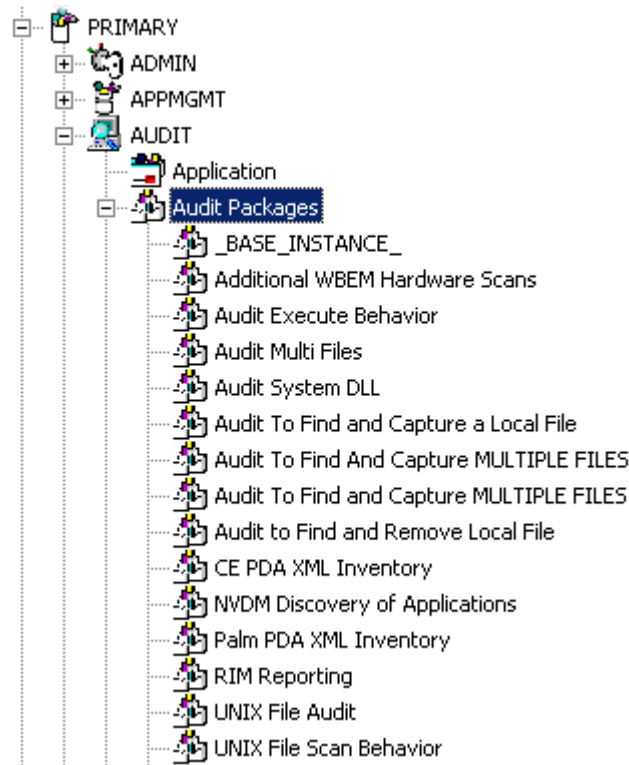
By viewing and deploying the sample audits provided by HP, system administrators will be able to create and use their own auditing packages.



If you are creating a WBEM Audit Package, be aware 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 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

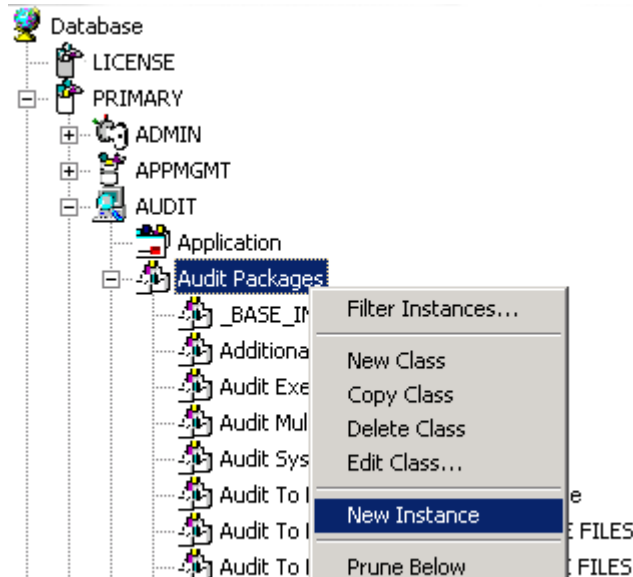
- 1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**. The CSDB Editor Security Information dialog box opens.
- 2 Type a User ID and, if necessary, a password, and then click **OK**. The CSDB Editor window opens.
 - ▶ The User ID, as shipped from HP, is **Admin** and password is **secret**. This may have been changed during installation. Check with your security administrator to obtain your own User ID and password, if necessary.
- 3 Double-click **PRIMARY**.
- 4 Expand the **AUDIT Domain**.
- 5 Double-click **Audit Packages (PACKAGE)** class.



As an example, we will create a new auditing package called Log Finder. This package will scan a user's computer for .log files, capture them, and return the results to the administrator.

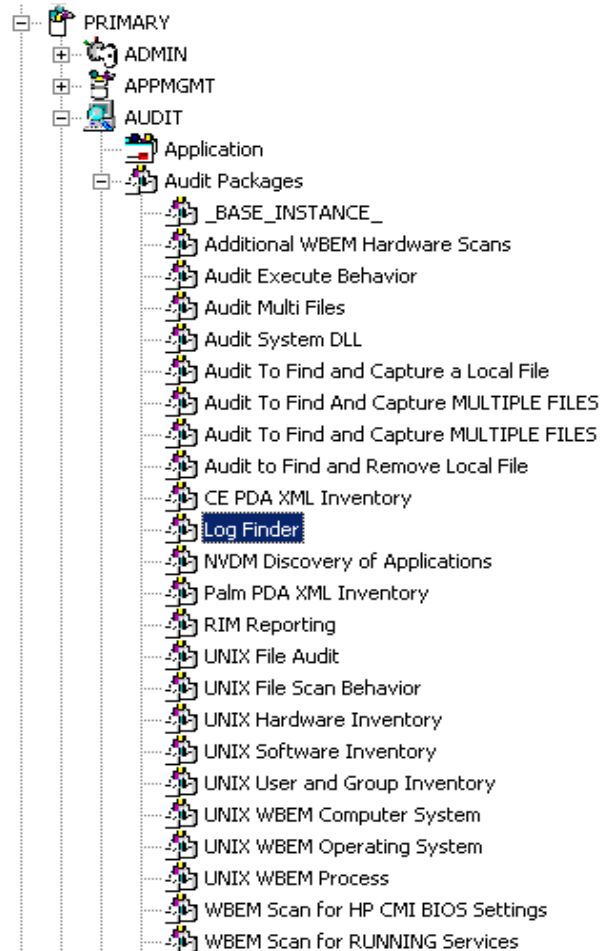
- 6 Right-click the **Audit Packages (PACKAGE)** Class.

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.

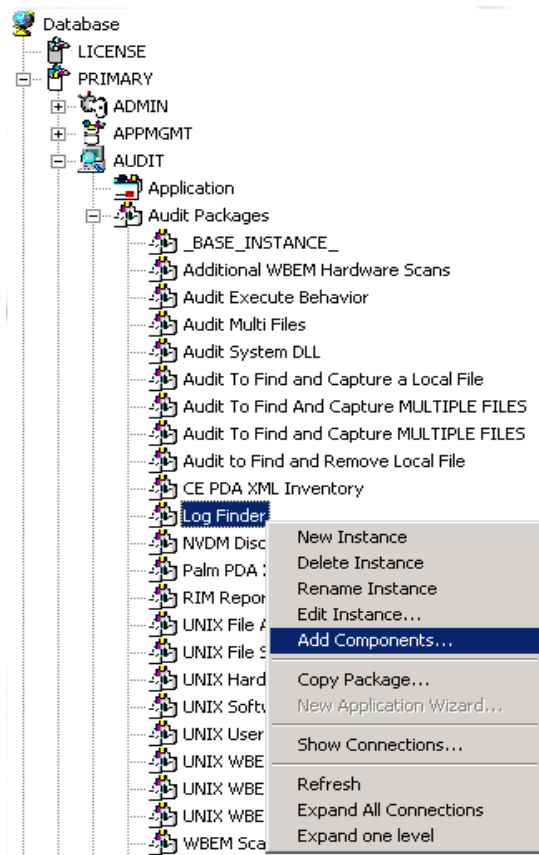
- 9 Enter a name for the Create a new Audit Packages (PACKAGE) instance name. This name appears in the title bar of the list view of the CSDB Editor window when the instance is selected and opened in the tree view.
- 10 Click **OK** to continue. The new Log Finder package is added to the AUDIT.PACKAGE Class.



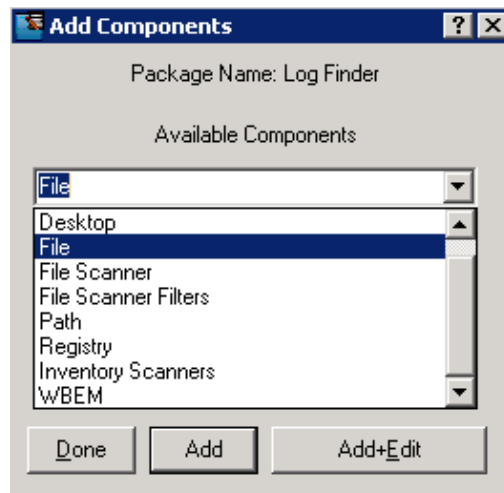
Once the Log Finder package is created, you will need to add its components.

To Add a Component to an Audit Package

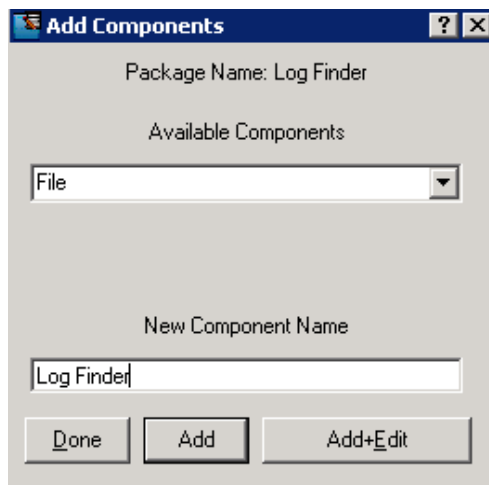
- 1 Right-click the **Log Finder** package. A shortcut menu opens.



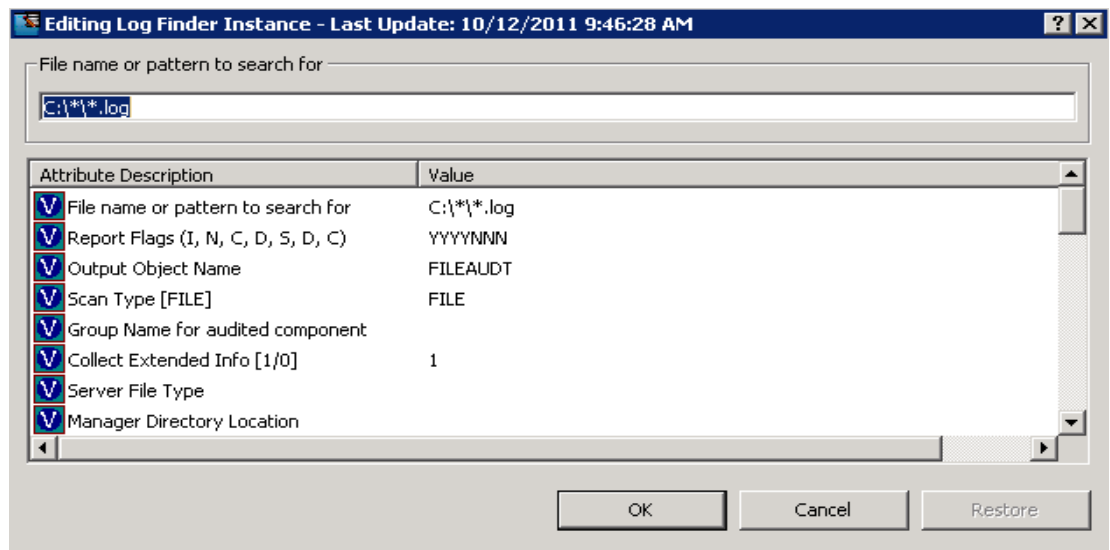
- 2 Select **Add Component** from the shortcut menu. The Add Components dialog box opens.
- 3 In the **Available Components** list, select **File**.



- 4 In the **New Component Name** box, enter the new component name.



- 5 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.

- 6 Click **OK** when you are done with your edit.
- 7 Click **Yes** to save your changes.

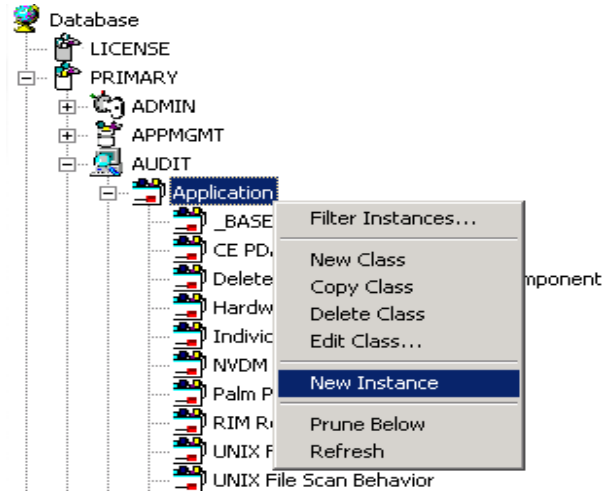
Next, you will need to create a **ZSERVICE** instance to contain the Log Finder package.

To Create a ZSERVICE Instance

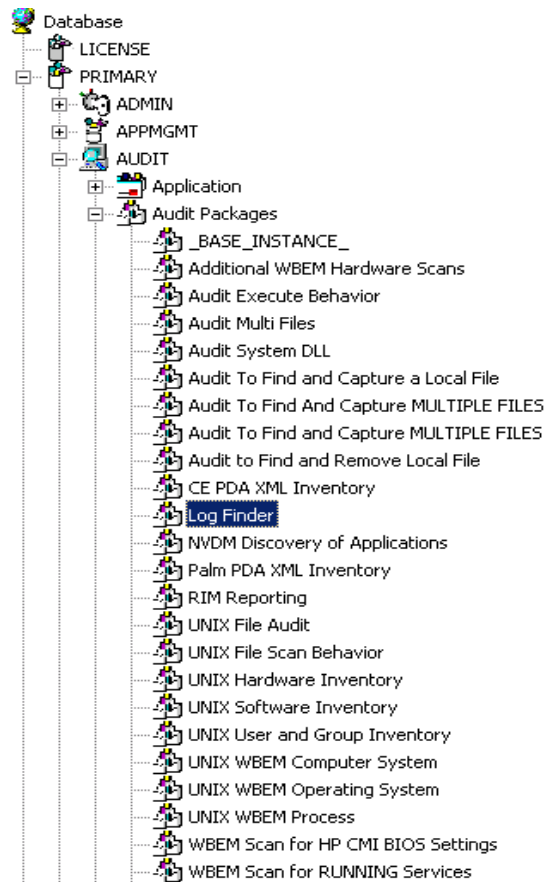


While working in the AUDIT Domain, note that the New Application Wizard is not available to connect a package to a service. You need to either copy an existing instance or create a new one.

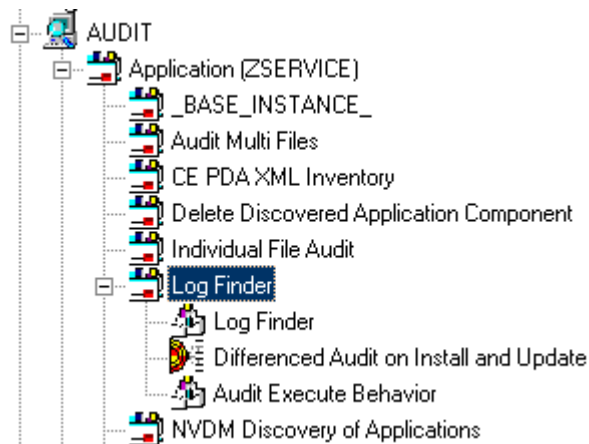
- 1 In the CSDB Editor, expand the AUDIT.ZSERVICE class in the tree view.
- 2 Right-click **Audit Application (ZSERVICE)** and a shortcut menu opens.



- 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 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 Application Self-Service Manager? Should the ZSVCNAME be changed? Should I enter additional information that may appear in the 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 Application Self-Service Manager, so we have changed the ZSVCMO attribute from mandatory to mandatory *and* optional. We would like the 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 Configuration Server's directory.


Figure 15 Log Finder ZSERVICE Attributes

Application class Log Finder Instance 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
ZSVCTYP	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]	
itc_ALWAYS_	Contains	AUDIT.PACKAGE.LOG_FINDER
dc_ALWAYS_	Contains	
dc_ALWAYS_	Contains	
dc_ALWAYS_	Contains	
dc_ALWAYS_	Contains	
ite_ALWAYS_	Contains	AUDIT.RIMOPTS.DIFF_INSTALL_UPD...
ite_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	Verified 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 CSDB Editor to connect and deploy the Log Finder audit service.

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

Figure 16 Log Finder in the Application Self-Service Manager

Name	Status	Compressed Size	Description	Mandatory	
Amortize	Available	n/a		<input type="radio"/>	
Drag & View	Available	2.51 MB		<input type="radio"/>	
GS-CALC	Available	n/a		<input type="radio"/>	
Log Finder					Size 6.67 KB
					Compressed Size 2.34 KB
Available					
Redbox Organizer	Available	n/a		<input type="radio"/>	
Sales Information	Available	n/a		<input type="radio"/>	
StratusPad	Available	n/a		<input type="radio"/>	

6 Configuring Timers for Audit Collection

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

The Scheduling (TIMER) Class

The Scheduling (TIMER) class enables the Client Automation administrator to set a timer on the Client Automation 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.

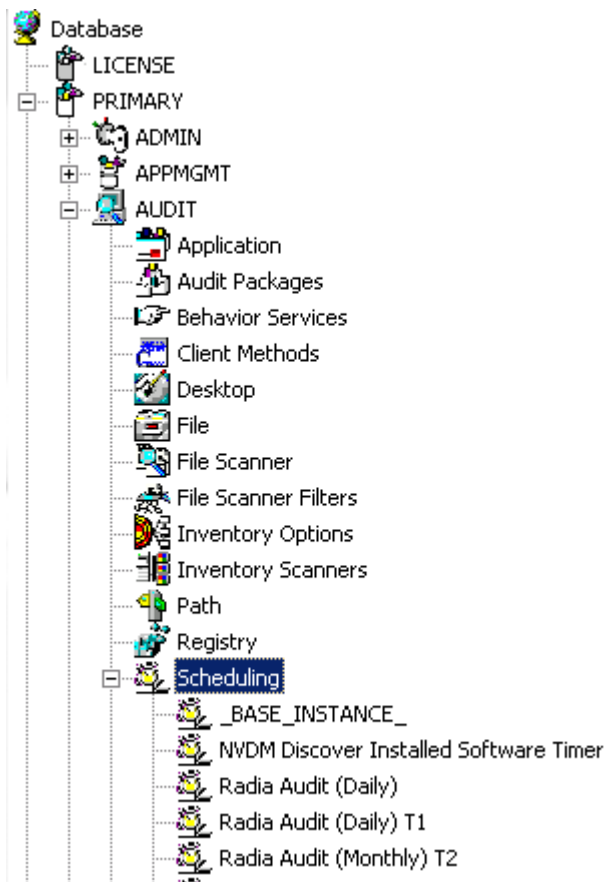
- ▶ As distributed by HP, the SOFTWARE Domain also contains a Scheduling (TIMER) class. Timers can be specified in instances of either of these Scheduling (TIMER) classes and can be connected to an Application (ZSERVICE) class instance in either the SOFTWARE or AUDIT Domain.

The following sample Timer packages are present within the AUDIT.Scheduling (TIMER) Class:

- **Daily** deploys a ZSERVICE everyday at the time specified.
- **Weekday** deploys a ZSERVICE on Mondays, Wednesdays, and Fridays at a specified time.
- **Weekly** deploys 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. See the *HP Client Automation Administrator Installation and 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 60.

Figure 17 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 Client Automation agent is installed with the Scheduler service. This service contains an executable timer component that executes any program on the end-user desktop when a timer expires.

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

The following table contains descriptions of the Scheduling (TIMER) class attributes:

Table 11 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 agent connect. The elements with a priority number less than the value of ZOBJPRI are deployed <i>before</i> the ZTIMEQ object. A value of 90 is inherited from the base instance and should not be changed.
ZSTOP	Used to assign timer conditions. Indicate true to cause resolution of the instance to be skipped. The timer is not deployed for end users. Leave <i>blank</i> 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.

Table 11 Scheduling (TIMER) Class

Attribute	Description
ZSCHTYPE	<p><i>Used only when ZSCHFREQ = PERIODIC.</i></p> <p>Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4 A.M. (non-peak) scheduled event that is sent to the Client Automation agent computer during the day. If it was not deferred, it would launch during the day instead of waiting until the next morning.</p> <p>Example 1:</p> <p>Suppose you create and deploy a timer with the ZSCHDEF = DAILY (&ZSYSDATE , 04 : 00 : 00).</p> <p>If ZSCHTYPE = IMMEDIATE and it is:</p> <ul style="list-style-type: none"> • Before 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. <p>If ZSCHTYPE = DEFERRED and it is:</p> <ul style="list-style-type: none"> • Before 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00. • After 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00. <p>Example 2:</p> <p>Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY (FRIDAY , 04 : 00 : 00)</p> <p>If ZSCHTYPE = IMMEDIATE and it is:</p> <ul style="list-style-type: none"> • Not Friday or Friday and before 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. <p>If ZSCHTYPE = DEFERRED and it is:</p> <ul style="list-style-type: none"> • 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	<p>This attribute indicates how often the timer should expire according to the frequency specified in the ZSCHDEF attribute.</p> <ul style="list-style-type: none"> • Once for a one-time expiration. • Periodic for a repeated expiration. • Random for random intervals.
ZRSCCMDL	<p>This attribute indicates the command line that is executed on the subscriber's computer when the timer expires.</p>
ZSVCOID	<p>Specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.</p>

Table 11 Scheduling (TIMER) Class

Attribute	Description
<code>_ALWAYS_</code>	Stores the connections to other instances
<code>NAME</code>	Friendly name for this instance
<code>APPSVC</code>	Application
<code>REQUEST</code>	Application request
<code>DOMAIN</code>	Server's domain name
<code>IPADDR</code>	Server's IP address/name
<code>SOCKET</code>	Server's socket number
<code>MGRNAME</code>	Server's name
<code>ZCREATE</code>	Scheduler <code>CREATE</code> method that runs on the Client Automation agent computer This value is inherited from the base instance and should not be changed.
<code>ZVERIFY</code>	Scheduler <code>VERIFY</code> method that runs on the Client Automation agent computer This value is inherited from the base instance and should not be changed.
<code>ZUPDATE</code>	Scheduler <code>UPDATE</code> method that runs on the agent computer This value is inherited from the base instance and should not be changed.
<code>ZDELETE</code>	Scheduler <code>DELETE</code> method that runs on the Client Automation agent computer This value is inherited from the base instance and should not be changed.
<code>RUNSYNC</code>	Sets the value of Yes or No for the synchronous timer execution. The default value is Yes .
<code>ZNOPING</code>	Controls the automatic sensing of a network connection between the Client Automation agent computer and the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. Use this attribute when there is a possibility that the Client Automation agent will not be able to connect with the Configuration Server. This attribute is especially useful for mobile users. Note: In order to use this attribute, you must add it to the <code>TIMER</code> class template.
<code>PINGDLAY</code>	Sets the amount of time between pings in milliseconds. The default setting is 2000 milliseconds .
<code>PINGCNT</code>	Sets the number of ping attempts to be made by the 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 CSDB, use the 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 *HP Client Automation Application Manager and Application Self-service Manager Installation and Configuration Guide*.

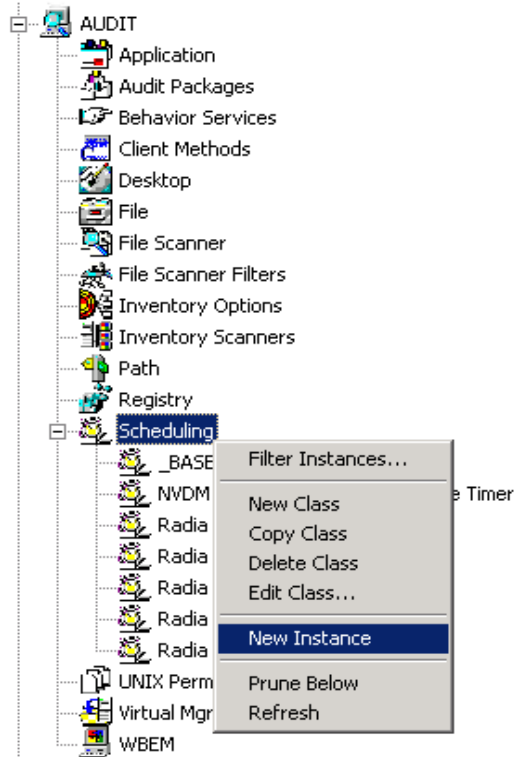
To Create a New Timer in the AUDIT Domain

- 1 From the Start menu, go to **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**. The Security Information dialog box opens.
- 2 Type a User ID and, if necessary, a password, and then click **OK**. The CSDB Editor window opens.

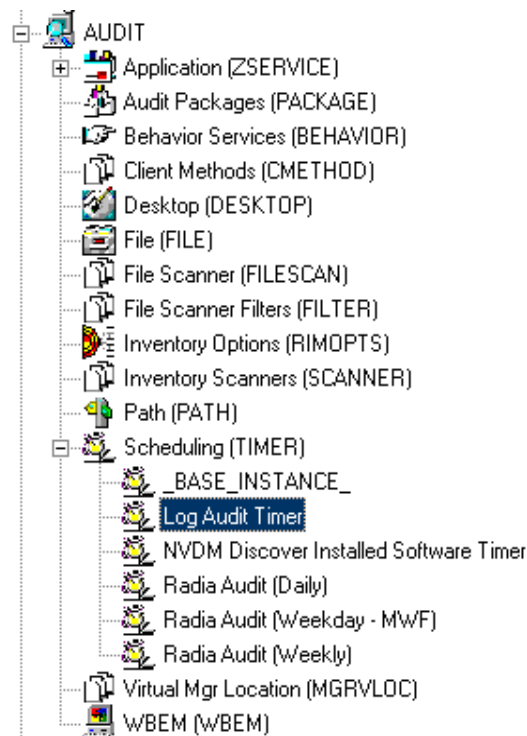


The User ID, as shipped from HP, is **Admin** and password is **secret**. This may have been changed during installation. Check with your security administrator to obtain your own User ID and password, if necessary.

- 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. See the *HP Client Automation Application Manager and Application Self-service Manager Installation and Configuration Guide* for more information on how to specify the Client Automation 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 11](#) on page 57 to help you determine the appropriate syntax.

Table 12 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 <i>Name of Weekday</i> * at 1:00 AM. The weekday must be specified in uppercase.
HOURLY	HOURLY (&ZSYSDATE, 08:41:00)	Hourly starting at 8:41 AM on the 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.ZMGRN
AME), dname=&(ZMASTER.ZDOMNAME), sname=&(ZSERVICE.ZOBJNAME)
```

► Execution causes Client Automation 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 Client Automation agent computer and the Configuration Server. The default is **Y**. Use this attribute when there is a possibility that the Client Automation agent will not be able to connect with the Configuration Server such as a mobile user.

See the *HP Client Automation Application Manager and Application Self-service Manager Installation and Configuration 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 Configuration Server.
- If ZNOPING = **n**, the Scheduler service will ping the Configuration Server.
 - If the Configuration Server is pinged successfully, the command in the ZRSCCMDL attribute is executed. The PENDING attribute in the Client Automation agent's ZTIMEQ object is then set to **N**. This indicates that the Scheduler service does not need to ping the Configuration Server again.
 - Set ZNOPING to **w** if you are specifying an end limit in the ZCHDEF attribute. The Scheduler pings the Configuration Server before executing the command. If the 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 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 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 Configuration Server.

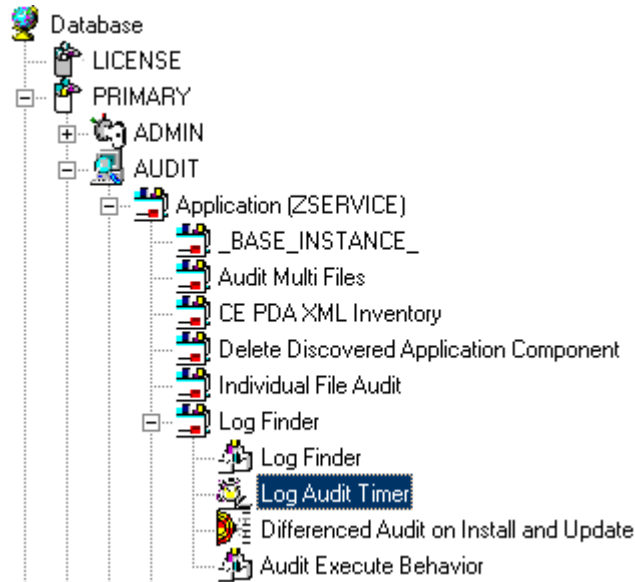
- If ZNOPING is set to **n**, PINGDLAY specifies the time in milliseconds between pings. The default is **2000**.
- If ZNOPING is set to **n**, PINGCNT specifies number of ping attempts. The default is **3** attempts.

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 Client Automation agent connects to the Configuration Server.

Use the CSDB Editor to connect the **Log Audit Timer** to the **Log Finder ZSERVICE** created earlier in this document.

Figure 18 Log Audit Timer Instance Connected to Log Finder Service



Then connect the `AUDIT.ZSERVICE.Log Finder` to a user or group of users in the **POLICY Domain**.

Figure 19 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 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`.

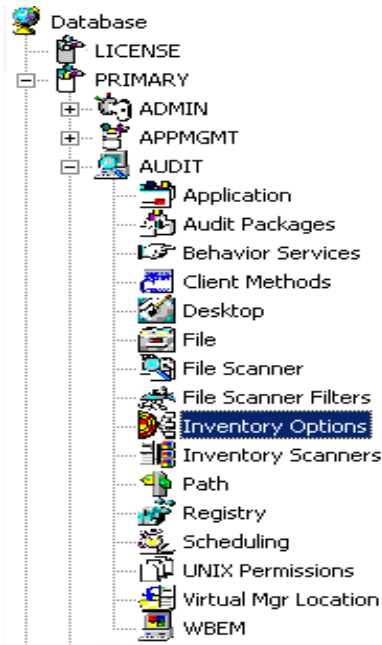
The easiest way to approach this issue is to manipulate how and when the audit actually executes. This can be accomplished by:

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

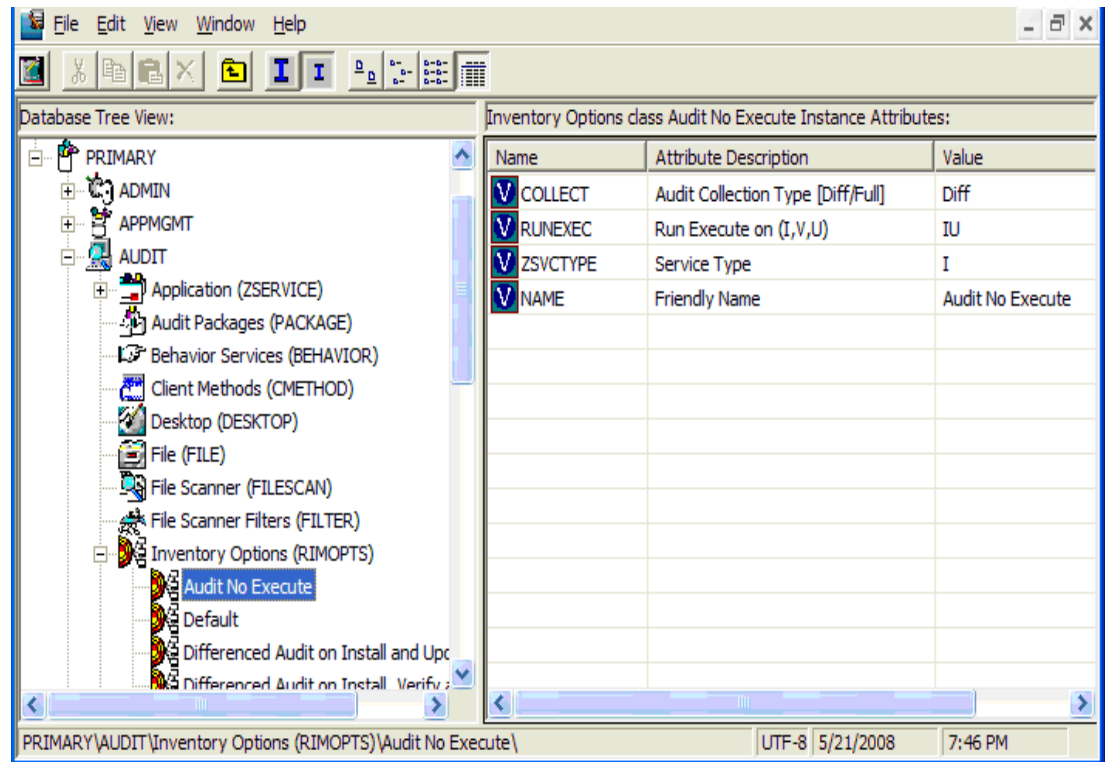
The following describes the steps necessary to customize RIMOPTS and update the embed tag to prevent audit execution during mandatory application processing.

To Customize the RIMOPTS Instance

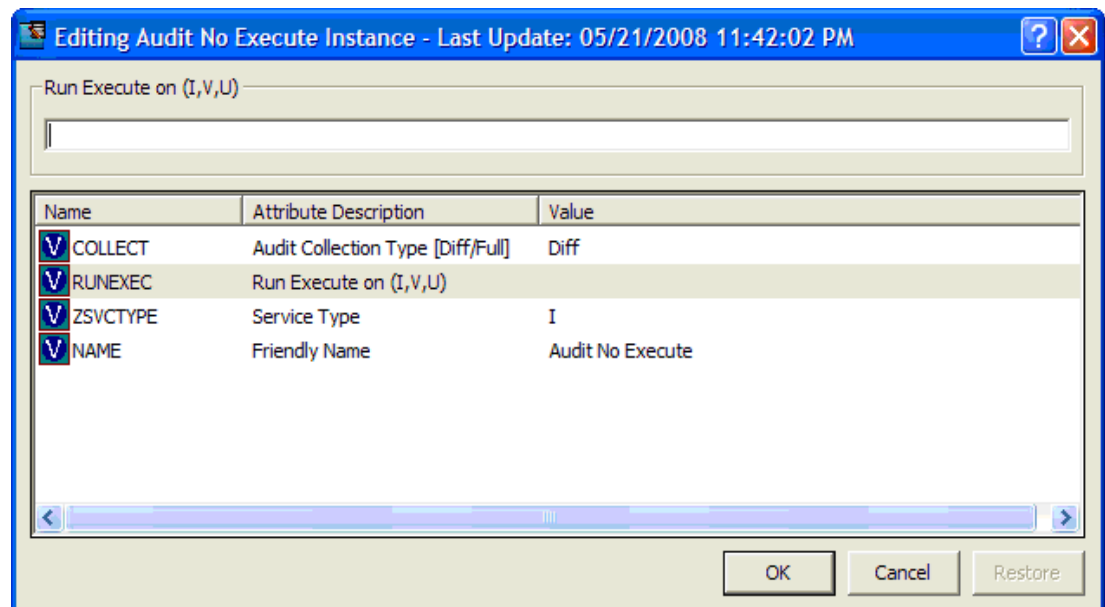
- 1 From the Start menu, select **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**. The CSDB Editor Security Information dialog box opens.
 - ▶ The User ID, as shipped from HP, is **Admin** and password is **secret**. This may have been changed during installation. Check with your 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 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 Audit No Execute instance.
- 5 Expand the **Inventory Options (RIMOPTS)** class and double-click the **Audit No Execute** instance.

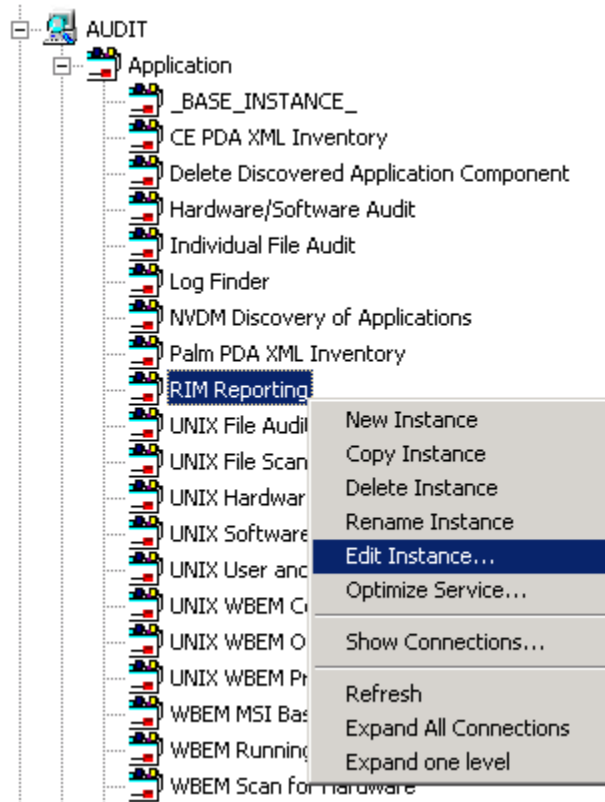


- 6 Double-click the **RUNEXEC** attribute in the list view to edit it. Remove any attribute information. This will ensure that the audit service will not run during the installation, verification, or update function.

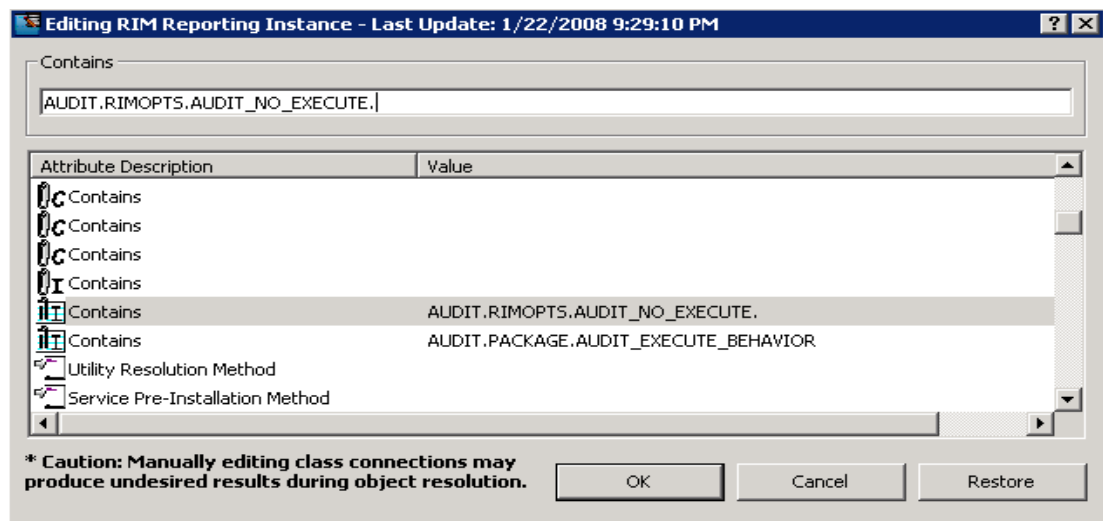


Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the RIM_REPORTING service.

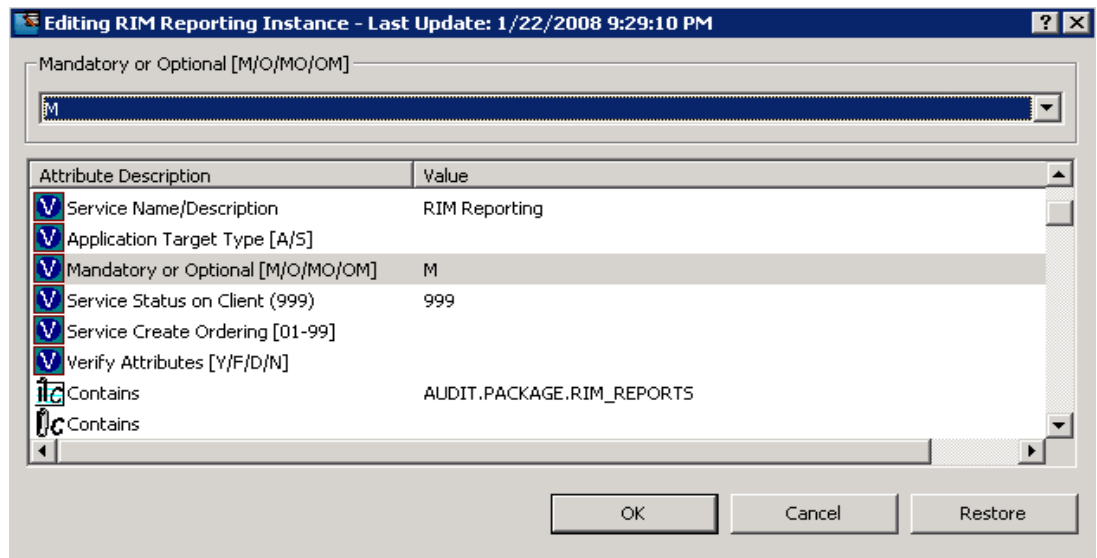
- 7 Right-click **RIM_REPORTING** Service in the AUDIT class.
- 8 Select **Edit Instance**.



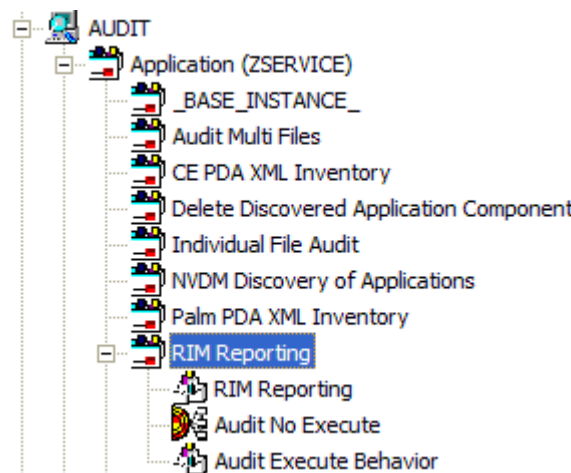
- 9 Locate the `_ALWAYS_ Contains` attribute with the value of `AUDIT.RIMOPTS.DIFF_INSTALL_UPDATE` and change it to a value of `AUDIT.RIMOPTS.AUDIT_NO_EXECUTE`.



- 10 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 Client Automation agent.



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



This completes the steps necessary to customize RIMOPTS and update the embed tag to prevent audit execution during mandatory application processing.

7 Viewing Inventory Reports

To view the Inventory reports:

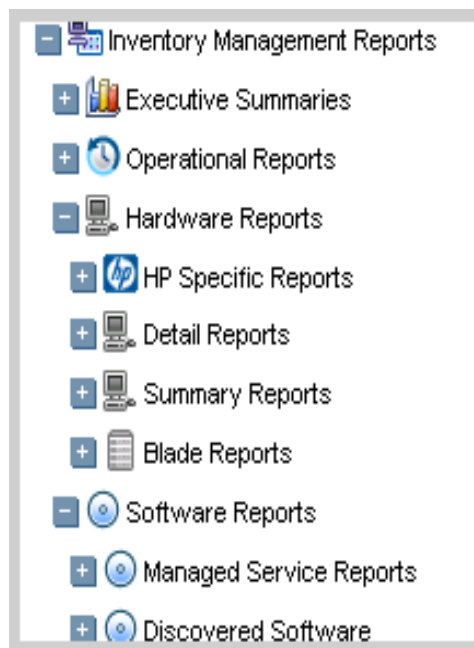
- 1 Go to your HPCA Console.
- 2 Click the **Reporting** tab.
- 3 Select the **Inventory Management Reports** under reporting views.

Reporting Views for Inventory Reports

There are different types of Inventory Management Reports:

- Executive Summaries
- Operational Reports
- Hardware Reports
- Software Reports
- Readiness Reports
- Power Utilization

Figure 20 Inventory Management Reports



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

Table 13 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
Summary Reports	Count by Summary Count by CPU Count by Memory Count by Operating System
Blade Reports	Blade by Racks Blade by Enclosures Managed Blades

Table 14 Software Reporting Views

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

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 Reporting Server determines Vista readiness based on the following criteria:

- CPU Speed
- Memory
- System Drive Total
- System Drive Free

See the 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 Vista's upgrade readiness of your devices. The Readiness Status and Additional Information columns contain information about the current level of readiness for each device.

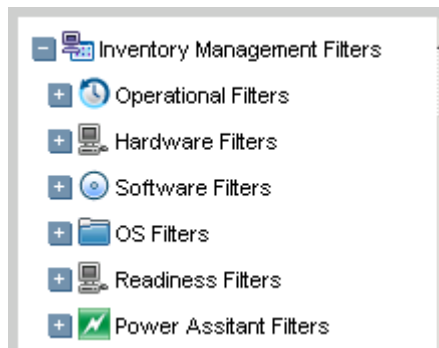
Filtering Inventory Reports with Reporting Server

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

Filter types include:

- Operational Filters
- Hardware Filters
- Software Filters
- OS Filters
- Readiness Filters
- Power Assistant Filters

Figure 21 Inventory Management Related Data Filters

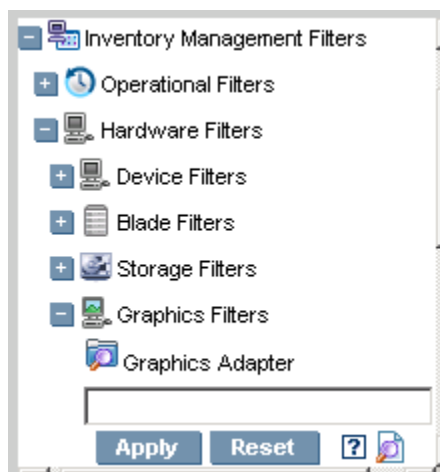


Expand each individual **Inventory Management** data filter to see the available filters you can apply to the current Reporting View.

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

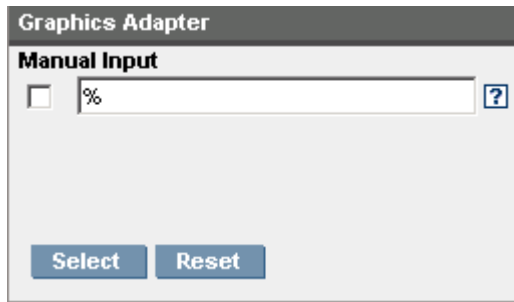
To get help when entering filter, point to the help icon and a tooltip specifies the syntax and gives examples.

Figure 22 Expand a Filter



Click the magnifying glass to open the filter lookup window.

Figure 23 Select the Filter



The image shows a dialog box titled "Graphics Adapter" with a sub-section "Manual Input". Inside this section, there is a checkbox on the left and a text input field on the right containing the character "%". A small blue question mark icon is located to the right of the input field. At the bottom of the dialog box, there are two buttons: "Select" and "Reset".

For additional information on creating filters and using the Reporting tab in general, see the *HP Client Automation Core and Satellite Enterprise Edition*.

A Detail and Summary Reporting Tables

Table 15 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	
				wVersion/Version	

Table 15 Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		wSoftwareElementID/ SoftwareSpec	
		wTargetOperatingSystem/ TargetOS	
		wManufacturer/Vendor	
		wCaption/Caption	
		wInstallDate/Install Date	
	Audited Applications	wCaption/Application Name	rNVD_Product
		mtime/Modified	
		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	

Table 15 Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		wRegisteredUser/Registered User	
		wOrganization/Organization	
		wSerialNumber/S/N	
		wSystemDirectory/Sys Dir	
		WtotalPageFileSpace/ PageFileSize (mb)	
	Hardware	manufacturererr/Manufacturer	rWin32_ComputerSystemPro duct
		_model/Model	RWin32_systemEnclosure
		_wSNTag/S/N	RWin32_Processor
		wManufacturer,wCurrentClo ckSpeed/Processor	RWin32_LogicalMemoryConf
		wTotalPhysicalMemory/ Physical Memory (MB)	rWin32_Computer System
		wSystemType/System	rWin32_Bios
		_wBios / Bios	
		_wKybd/keyboard	RWin32_Keyboard
		_wMouse/Mouse	rWin32_PointingDevice
		_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	

Table 15 Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		Sysdrv_total/ Sys Drive Size (MB)	
		Sysdrv_free/Sys Drive Free (MB)	
	Software (AGENT)	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 (#)	
		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	

Table 15 Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
	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)	

Table 15 Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
	PDA Installed Products	Show-pda-inst-prod.tsp	RCIM_OperatingSystem
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	

Table 15 Inventory Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		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 16 General Reporting – Detailed Reports

Action	Display 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	

Table 16 General Reporting – Detailed Reports

Action	Display Table Title	Columns Queried/Display Name	Tables Queried
		edmllog/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	
		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 (#)	

Table 16 General Reporting – Detailed Reports

Action	Display Table Title	Columns Queried/Display Name	Tables Queried
		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	
		nfy_addr/Address	
		nfy_port/Port	
		nfy_maxretry/Max (#)	
		nfy_delay/Delay (s)	

Table 16 General Reporting - Detailed Reports

Action	Display Table Title	Columns Queried/Display Name	Tables Queried
		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 17 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	

Table 17 History Reporting – Detailed Reports

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		svc_count/Services (#)	
		ver_error/Vers Error	
		rsrc_count/Files (#)	
		rsrc_error/File Err	
		rsrc_active/Files Active (#)	
		rsrc_active_size/Files Active (Sz)	
		rsrc_inactive/Files Inactive (#)	
		rsrc_inactive_size/Files Inactive (Sz)	
		reason/Reason	

Table 18 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

Table 18 Summary Reporting

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		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
		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	

Table 18 Summary Reporting

Action	Displayed Table Title	Columns Queried/Display Name	Tables Queried
		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 19 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	

Table 19 Inventory Reporting – Multicast Detail Reporting

Action	Displayed Table Title	Columns Queried /Display Name	Tables Queried
		wNpktsDrp	
		wNpktsRcv	
		wServiceID/Service	
		wSourceID	
		wSourceType/Source Type	
		wStartTime/Transmit Start	

We appreciate your feedback!

If an email client is configured on this system, click

[Send Email](#)

If no email client is available, copy the following information to a new message in a web mail client and send the message to **docfeedback@hp.com**.

Product name and version: HP Client Automation 8.10

Document title:

Feedback:

