Radia Client Automation Enterprise Inventory Manager

For the Linux operating system

Software Version: 9.00

Reference Guide

Document Release Date: April 2013

Software Release Date: April 2013

PERSISTENT

Legal Notices

Warranty

The only warranties for products and services are set forth in the express license or service agreements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Persistent Systems 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 Persistent Systems or its licensors required for possession, use or copying. No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Persistent Systems.

Copyright Notice

Trademark Notices

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).

This product includes software written by Daniel Stenberg (daniel@haxx.se).

This product includes OVAL language maintained by The MITRE Corporation (oval@mitre.org).

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://support.persistentsys.com/

This site requires that you register for a Persistent Passport and sign in. Register online at the above address.

For more details, contact your Persistent sales representative.

Support

Persistent Software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Submit enhancement requests online
- Download software patches
- Look up Persistent support contacts
- Enter into discussions with other software customers
- Research and register for software training

To access the Self-solve knowledge base, visit the Persistent Support home page.

Note: Most of the support areas require that you register as a Persistent Support user and sign in. Many also require an active support contract. More information about support access levels can be found on the Persistent Support site.

To register for a Persistent Support ID, go to: Persistent Support Registration.

Contents

Chapter 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 Radia 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 RCA server for later reporting.

For more information on reporting, see the *Radia Client Automation Enterprise User Guide*.

Terminology

agent computer

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

agent

The **Radia Client Automation Agen**t is the RCA software component that is installed on the end user's desktop computer. There are RCA agents for the Application Manager, the Application Self-service Manager, the Inventory Manager, the Patch Manager, and the OS Manager.

clean machine

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

Common Information Model (CIM)

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

Messaging Server

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

Reporting Server

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

subscriber

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

Web-Based Enterprise Management (WBEM)

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

RCA Prerequisites

The Inventory Manager requires the following RCA components:

- RCA Server
- Client Automation agent
  - Application Manager with Inventory Manager feature
  - Application Self-Service Manager (optional)
- RCA Administrator CSDB Editor. This is installed as part of the RCA Administrator. For more information on RCA Admin CSDB Editor, see the *Radia Client Automation Enterprise CSDB Editor Online Help*.

Necessary Skills

The following skills are required to use the Inventory Manager.

With HPCA Products

This document assumes that the reader is familiar with the CSDB and with administering RCA using the Admin CSDB Editor. For more information, see the *Radia Client Automation Enterprise CSDB Editor Online Help*.

With Web-Based Enterprise Management

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

`http://www.dmtf.org/standards/wbem`

Inventory Manager Technology

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

Web-Based Enterprise Management (WBEM)

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

Common Information Model (CIM)

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

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

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

RCA and WBEM

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

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

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

Chapter 2

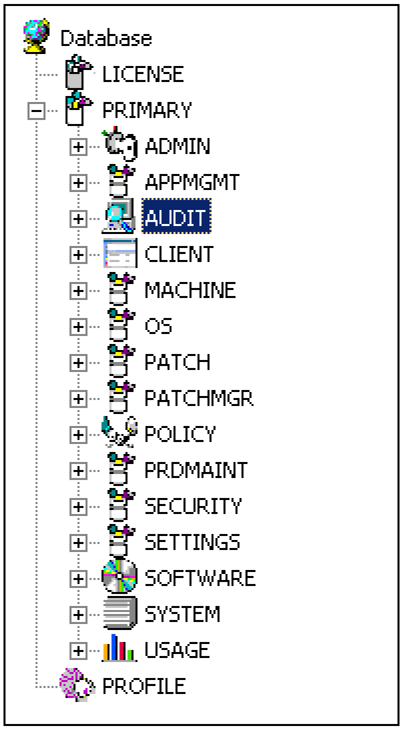
The AUDIT Domain

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

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

> **Note:** The following figures and instructions use the Admin CSDB Editor, which is available for 32-bit Windows platforms. For more information, see the *Radia Client Automation Enterprise CSDB Editor Online Help*.

PRIMARY.AUDIT Domain



AUDIT Domain Defined

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

Audit Domain

Class	Description
Audit Application (ZSERVICE)	Sample services distributed with the Inventory Manager. The AUDIT.ZSERVICE instance is connected to a policy instance. A policy instance can be an instance of the Users, Departments, or Workgroups class. It can also be a customer-defined class within the POLICY Domain. Each of the sample ZSERVICE Classes is connected to the PACKAGE instances.
Audit Packages (PACKAGE)	Defines what information to collect and then what actions to take. These packages would contain various audit components. A good example is an audit of running services on a desktop. The AUDIT.ZSERVICE instance must contain a connection to an AUDIT.PACKAGE instance.
Behavior Services (BEHAVIOR)	Defines instances that enable the execution of auditing on the agent. Normally, there is no need to add or modify instances in this class.
Client Methods (CMETHOD)	Used to configure method points for Tcl inventory scans. The base instance of the SCANNER Class is connected to the CMETHOD.INV_FULL instance. This instance can be used for all inventory scans defined in the SCANNER Class.
Desktop (DESKTOP)	This class is reserved for future use.
File (FILE)	Defines file scans, such as auditing system executables.
File Scanner (FILESCAN)	Persistent component class used to configure an inventory scan. Adding File Scanner components to an audit package creates instances of the FILESCAN class.
File Scanner Filters (FILTER)	Persistent component class used to configure an inventory scan. Adding File Scanner Filters components to an audit package creates instances of the FILTER class.
Inventory Options (RIMOPTS)	Contains the attributes that offer options to control an inventory management task.
Inventory Scanners (SCANNER)	Persistent component class that is used to configure an inventory scan. Create instances of the SCANNER class by adding Inventory Scanners components to an audit package.
Path (PATH)	Stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class.
Registry (REGISTRY)	Uses WMI to obtain a Registry scan of a Windows machine. Create instances of the REGISTRY class to run scans of the Windows Registry and obtain a Registry Scan report. For more information, see the *Registry Class* topics.
Scheduling (TIMER)	Contains the instances that enable the RCA administrator to set a timer on end users' computers. One or multiple auditing services can be processed whenever the timer expires.

Class	Description
UNIX Permissions (UNIXPERM)	Contains UNIX® file permission information.
Virtual Mgr Location (MGRVLOC)	Used to specify the initial path for files being transferred to the Configuration Server during a FILE audit.
WBEM (WBEM)	Contains instances that define Inventory Manager scans of WMI classes. These can include any class in the WMI database such as Win32_ Services.

FILTER Instances

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

RIMOPTS Class

The RIMOPTS Class is also known as the Inventory Options Class. This class contains the attributes that control an inventory management task. The following table RIMOPTS Class describes these attributes.

RIMOPTS Class

Attribute	Usage
COLLECT	Audit Collection Type by selecting **Diff** or **Full** • Select **Diff** to report the difference between the previous information collected for the service and the information collected during the current agent audit. This is the default setting. **Note:** The first or initial scan of the DIFF setting will be a FULL scan as defined below. All subsequent scans will then be differenced unless the administrator changes the setting to FULL. • Select **Full** to report the information collected for the service during the current agent connect process without differencing against the previous collection for that service.
RUNEXEC	Indicates what actions the Inventory Manager will take upon connection: • Select **I** to invoke collection of information when the service is installed • Select **U** to invoke collection of information when the service is updated. • Select **V** to invoke collection of information when the service is verified. The default settings are **I** and **U**.
ZSVCTYPE	Contains code that is used internally by the 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 Admin CSDB Editor.

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

Prior to beginning any tasks using the Inventory Manager, you must enable the drag-and-drop feature for the newly created RIMOPTS Class instances. For more information about editing instances, see the *Radia Client Automation Enterprise CSDB Editor Online Help*.

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

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

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

Chapter 3

Software and Hardware Auditing

CIM Schema and Inventory Collection

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

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

For a description of the CIM schema classes used, see the following table CIM Classes.

CIM classes

CIM Class	Description
CIM_SCSIController	Subclass of the CIM_Controller used to represent SCSI controllers.
CIM_ResidesOnExtent	Subclass of CIM_Dependancy. This is an association between the logical volume and the file system on the logical volume.
CIM_Processor	Used to represent computer processor information.
CIM_ParallelController	Subclass of CIM_Controller used to represent parallel controllers.
CIM_NFS	Used to represent general information about NFS mounted file systems.
CIM_MediaPresent	Used to represent relationship with the MediaAccessDevice. Represents logical volume or volume group and one of the disks it resides on.
CIM_ LogicalDiskBasedOnVolume	Subclass of LogicalDiskBasedOnExtent used to represent the relationship between logical volume and its volume group.
CIM_LogicalDisk	Used to represent general information about the logical volume.
CIM_IDEController	Subclass of CIM_Controller used to represent IDE controllers, including ATA and ATAPI controllers.
CIM_EthernetAdapter	Used to represent capabilities of the Ethernet card.
CIM_DiskDrive	Subclass of CIM_MediaAccessDevice, includes all hard disk drives, non-removable and removable. Models the reader/writer properties of disk drives.

CIM Class	Description
CIM_Directory	Used for exported directory.
CIM_DVDDrive	Subclass of CIM_MediaAccessDevice includes all of the types of DVD reader and writer drives.
CIM_CDROMDrive	Subclass of CIM_MediaAccessDevice includes CDROM reader and writer drives.
CIM_Service	Used to represent general information about NFS client/server service.
CIM_SCSIInterface	Subclass of CIM_ControlledBy. Represents unique data from the relationship between the controller and the device.
CIM_UnixLocalFileSystem	Used to represent UNIX specific information about the local file system.
CIM_ UnixComputerFileSystem	Used to represent general information about the computer.
CIM_StorageVolume	Used to represent the hand-off point between providers or the result of a redundancy.
CIM_UnixOperatingSystem	Used to represent general information about the UNIX operating system. General information about the volume groups.
CIM_SoftwareElement	Used to represent the SVR4 packages or filesets.
CIM_Export	Used to represent an association between a LocalFileSystem and its directories indicating that the specified directories are available for mount. When exporting an entire FileSystem, the directory should reference the topmost directory of the FileSystem.

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

http://www.dmtf.org/

Auditing Types

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

The Inventory Manager allows for three types of audits:

- File auditing
- WBEM auditing
- Hardware auditing

File Auditing

AUDIT.FILE

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

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

AUDIT.FILE Class Instances

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

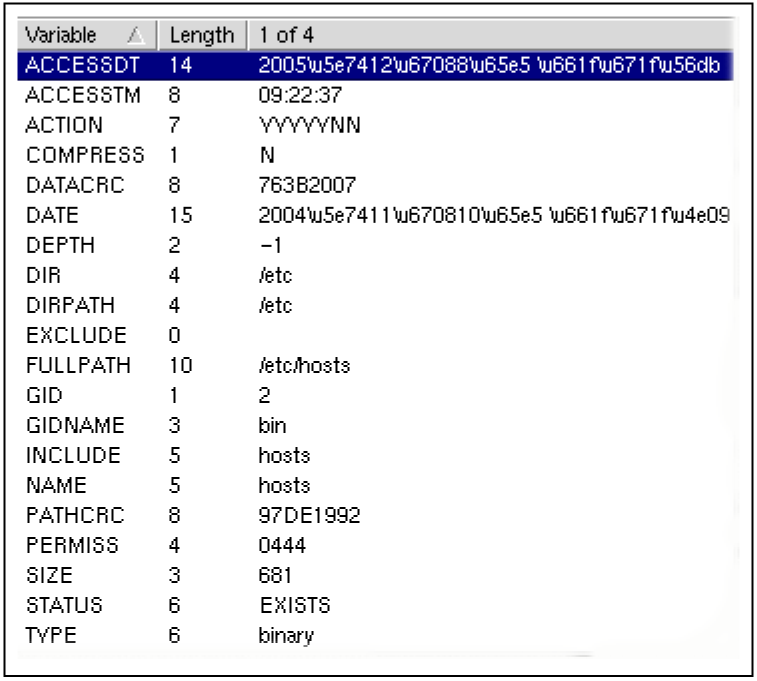
Attribute	Description Examples
	- **Custom** means to execute the method indicated in the CUSTOM attribute. YYYYNYN – Report whenever encountered and delete the files. NNYYNNN – Report when changed or deleted and take no action. NYYNYYN – Report when the files are new or changed. Then send and delete the files.
OUTPUT	Output object name.
TYPE	Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM. File.
GROUP	Optional way to identify a set of scan results. This maybe useful for querying and reporting on the audited files from the database where audit results can be stored. Games, MPEGs.
ZVERINFO	Collect extended information. - Set the value to **1** to collect more information for a file. - Set the value to **0** to not collect more information. In order for this data to be collected, the associated attribute must exist in the AUDIT.FILE class template. You can limit the scan to only those files that have some particular values in their extended information. You do so by supplying a value (either 1 or 0) for any of the associated attributes in an AUDIT.FILE instance. This causes the scan to be filtered. Only those files whose extended information data element contains the value you specify in its associated attribute will be scanned. Extended file information consists of one or more of the following data elements. The associated attribute name for the data element is in parentheses: (VENDOR) The seller of the file/product (PRODUCT) The name of the item for which the file is a part. (PRODVERS) The version of the product which the file is a part. (ORGNAME) The name of the organization. (INTERNAL) The internal data element encoded in the file.

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

	Description
Attribute	**Examples**
ZRSCSRC	Resource Source, i.e. Publisher.
ZINIT	Not applicable at this time.
NAME	Not applicable at this time.
LOCATION	Not applicable at this time.
ZMD5INFO	Set to **Y** to collect MD5 info. This is a 32-character value that can be used to uniquely identify a file based on its contents.

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

FILEPREV object created with RIMFSCAN



Variable	Length	1 of 4
ACCESSDT	14	2005\u5e7412\u67088\u65e5 \u661f\u671f\u56db
ACCESSTM	8	09:22:37
ACTION	7	YYYYYNN
COMPRESS	1	N
DATACRC	8	763B2007
DATE	15	2004\u5e7411\u670810\u65e5 \u661f\u671f\u4e09
DEPTH	2	-1
DIR	4	/etc
DIRPATH	4	/etc
EXCLUDE	0	
FULLPATH	10	/etc/hosts
GID	1	2
GIDNAME	3	bin
INCLUDE	5	hosts
NAME	5	hosts
PATHCRC	8	97DE1992
PERMISS	4	0444
SIZE	3	681
STATUS	6	EXISTS
TYPE	6	binary

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

FILEPREV Object

Attribute	Description
ACTION	Action flags. First four flags determine when to report. Y – ignored Y – New file Y – File changed since last scan

Attribute	Description
	Y – Ignored Last three flags control action to be taken. Y – send the file to RCS Y – ignored Y – ignored
ACCESSDT	The date of the most recent access of this file.
ACCESSTM	The time of the most recent access of this file.
COMPRESS	Compression setting.
DATACRC	Data CRC
DATE	The date of the most recent modification to this file.
DIR	System drive location of the file.
DIRPATH	The directory path of the file.
EXCLUDE	Parameter to exclude.
FULLPATH	Fully qualified path and file name of the file.
GID	Unix group ID of file owner.
GIDNAME	Unix group name of file owner.
INCLUDE	Parameter to include.
NAME	File name.
PATHCRC	A unique number that indicates the CRC path used for differencing.
PERMISS	4-digit octal value for file permissions.
SIZE	File size in bytes.
TIME	The time of the most recent modification to this file.
TYPE	File type. Can be directory, LINK, or binary.
UID	UNIX ID of file owner.
UIDNAME	Username of the file owner.
ZOBJDATE	Date
ZOBJPCLAS	Class
ZOBJCID	Object Child ID

Attribute	Description
ZOBJPNAM	Unique Name
ZOBJTIME	Time
ZRSCVLOC	Location

Audit.FILESCAN

> **Note:** UNIX file auditing using `filescan.tkd` is supported for legacy purposes only. New installations of the Inventory Manager should use RIMFSCAN and RIMDIFF, described in the section AUDIT.FILE, above.

> **Note:** Resolution of the `filescan.tkd` service is not supported over SSL.

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

See "Inventory Scan Results" on page 38 for more information on the `filescan.tkd` methods.

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

AUDIT.FILESCAN Class Instances

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

WBEM Auditing

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

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

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

AUDIT.WBEM Instance

Attribute Name	Description
ACTION	RIMDIFF method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect. - **Y** configures RIMDIFF to perform the reporting action. - **N** configures RIMDIFF to not perform the reporting action. The first four flags determine when to report that the WBEM namespace instance was found: Report on: **Initial, New, Changed, Deleted, Scan, Delete, Custom** - **Initial** means that the file was found during the first scan of the agent computer. - **New** means that the file was found during the current scan. The file was not present during the previous scan. - **Changed** means that the file was present during the previous scan and is different from the file found during the current scan. - **Deleted** means that the file was found during the previous scan. The file is not present for the current scan. - **Scan** means that the file was found during the current scan. - **Delete** means that the file was found during the previous scan. The file is not present for the current scan. - **Custom** means that the file was found during a custom scan. The last three flags are not applicable to WBEM audits.
NAMESPACE	Name of the WBEM namespace to query or HARDWARE.
CLASS	Name of the WBEM Class to query or HARDWARE.
PROPERTY	Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.
CNDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.
NAME	Friendly name for this instance. This name will appear in the Admin CSDB Editor tree view to identify this instance.

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

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

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

WBEM object attributes in the agent

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

Manual Scanning Using RIMWBEM

Note: Start CIM Server before using the RIMWBEM.

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

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

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

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

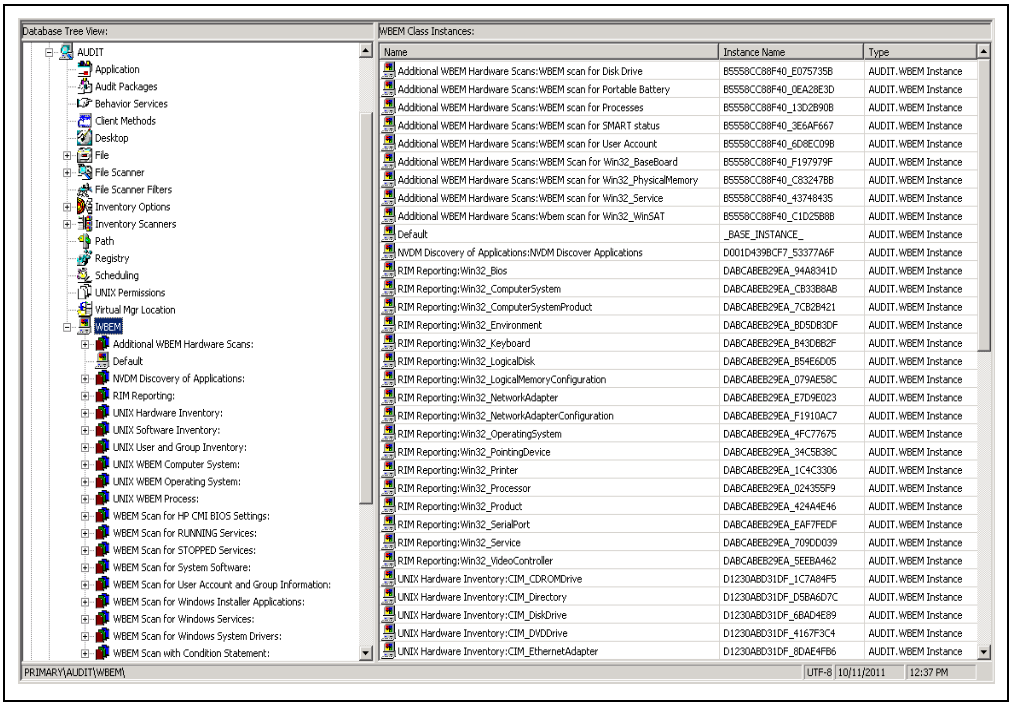
NVDCIM.TKD

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

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

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

AUDIT.WBEM Class instances


Database Tree View:
AUDIT
Application
Audit Packages
Behavior Services
Client Methods
Desktop
File
File Scanner
File Scanner Filters
Inventory Options
Inventory Scanners
Path
Registry
Scheduling
UNIX Permissions
Virtual Mgr Location
WBEM
Additional WBEM Hardware Scans:
Default
NVDM Discovery of Applications:
RIM Reporting:
UNIX Hardware Inventory:
UNIX Software Inventory:
UNIX User and Group Inventory:
UNIX WBEM Computer System:
UNIX WBEM Operating System:
UNIX WBEM Process:
WBEM Scan for HP CMI BIOS Settings:
WBEM Scan for RUNNING Services:
WBEM Scan for STOPPED Services:
WBEM Scan for System Software:
WBEM Scan for User Account and Group Information:
WBEM Scan for Windows Installer Applications:
WBEM Scan for Windows Services:
WBEM Scan for Windows System Drivers:
WBEM Scan with Condition Statement:
PRIMARY\AUDIT\WBEM\
WBEM Class Instances:
UTF-8
10/11/2011
12:37 PM

Name	Instance Name	Type
Additional WBEM Hardware Scans:WBEM scan for Disk Drive	B5558CC88F40_E075735B	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM scan for Portable Battery	B5558CC88F40_0EA28E3D	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM scan for Processes	B5558CC88F40_13D2B90B	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM scan for SMART status	B5558CC88F40_3E6AF667	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM scan for User Account	B5558CC88F40_6D8EC09B	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM Scan for Win32_BaseBoard	B5558CC88F40_F197979F	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM scan for Win32_PhysicalMemory	B5558CC88F40_C83247BB	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:WBEM scan for Win32_Service	B5558CC88F40_43748435	AUDIT.WBEM Instance
Additional WBEM Hardware Scans:Wbem scan for Win32_WinSAT	B5558CC88F40_C1D25B8B	AUDIT.WBEM Instance
Default	_BASE_INSTANCE_	AUDIT.WBEM Instance
NVDM Discovery of Applications:NVDM Discover Applications	D001D439BCF7_53377A6F	AUDIT.WBEM Instance
RIM Reporting:Win32_Bios	DABCABEB29EA_94A8341D	AUDIT.WBEM Instance
RIM Reporting:Win32_ComputerSystem	DABCABEB29EA_CB33B8AB	AUDIT.WBEM Instance
RIM Reporting:Win32_ComputerSystemProduct	DABCABEB29EA_7CB2B421	AUDIT.WBEM Instance
RIM Reporting:Win32_Environment	DABCABEB29EA_BD5DB3DF	AUDIT.WBEM Instance
RIM Reporting:Win32_Keyboard	DABCABEB29EA_B43DBB2F	AUDIT.WBEM Instance
RIM Reporting:Win32_LogicalDisk	DABCABEB29EA_B54E6D05	AUDIT.WBEM Instance
RIM Reporting:Win32_LogicalMemoryConfiguration	DABCABEB29EA_079AE58C	AUDIT.WBEM Instance
RIM Reporting:Win32_NetworkAdapter	DABCABEB29EA_E7D9E023	AUDIT.WBEM Instance
RIM Reporting:Win32_NetworkAdapterConfiguration	DABCABEB29EA_F1910AC7	AUDIT.WBEM Instance
RIM Reporting:Win32_OperatingSystem	DABCABEB29EA_4FC77675	AUDIT.WBEM Instance
RIM Reporting:Win32_PointingDevice	DABCABEB29EA_34C5B38C	AUDIT.WBEM Instance
RIM Reporting:Win32_Printer	DABCABEB29EA_1C4C3306	AUDIT.WBEM Instance
RIM Reporting:Win32_Processor	DABCABEB29EA_024355F9	AUDIT.WBEM Instance
RIM Reporting:Win32_Product	DABCABEB29EA_424A4E46	AUDIT.WBEM Instance
RIM Reporting:Win32_SerialPort	DABCABEB29EA_EAF7FEDF	AUDIT.WBEM Instance
RIM Reporting:Win32_Service	DABCABEB29EA_709DD039	AUDIT.WBEM Instance
RIM Reporting:Win32_VideoController	DABCABEB29EA_5EEBA462	AUDIT.WBEM Instance
UNIX Hardware Inventory:CIM_CDROMDrive	D1230ABD31DF_1C7A84F5	AUDIT.WBEM Instance
UNIX Hardware Inventory:CIM_Directory	D1230ABD31DF_D5BA6D7C	AUDIT.WBEM Instance
UNIX Hardware Inventory:CIM_DiskDrive	D1230ABD31DF_6BAD4E89	AUDIT.WBEM Instance
UNIX Hardware Inventory:CIM_DVDDrive	D1230ABD31DF_4167F3C4	AUDIT.WBEM Instance
UNIX Hardware Inventory:CIM_EthernetAdapter	D1230ABD31DF_8DAE4FB6	AUDIT.WBEM Instance

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

AUDIT.WBEM Instance

Attribute	Description
ACTION	The `filescan.tkd` method performs actions on the WBEM namespaces (s) instances discovered on the user's computer during the agent connect. • Y configures `filescan.tkd` to perform the reporting action. • N configures `filescan.tkd` to not perform the reporting action. The first four flags determine when to report that the WBEM namespace instance was found:

Attribute	Description
	Report on: `Initial, New, Changed, Deleted` • `Initial` means that the file was found during the first scan of the agent computer. • `New` means that the file was found during the current scan. The file was not present during the previous scan. • `Changed` means that the file was present during the previous scan and is different from the file found during the current scan. • `Deleted` means that the file was found during the previous scan. The file is not present for the current scan. The last three flags are not applicable to WBEM audits.
NAMESPACE	Name of the WBEM namespace to query or HARDWARE.
CLASS	Name of the WBEM Class to query or HARDWARE.
PROPERTY	Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported.
CNDITION	An optional condition to narrow results of an audit.
OUTPUT	This is the name of the object to send to the Configuration Server.
TYPE	Indicates that WBEM scan is to be employed for this audit package.
NAME	Friendly name for this instance. This name will appear in the Admin CSDB Editor tree view to identify this instance.

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

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

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

WBEM object attributes

Attribute	Description
ZOBJCID	Object child ID.
ZOBJCLAS	The targeted class for the audit such as ZRSOURCE or ZSERVICE.
ZOBJCNUM	Number of children under current instance.
ZOBJCRC	The CRC of all persistent and transient objects under the current node.

Attribute	Description
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.
ZUNUSED1	For future use.

WBEM Object Processing

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

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

For more information, see the *Radia 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 Configuration Server database:

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

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

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

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

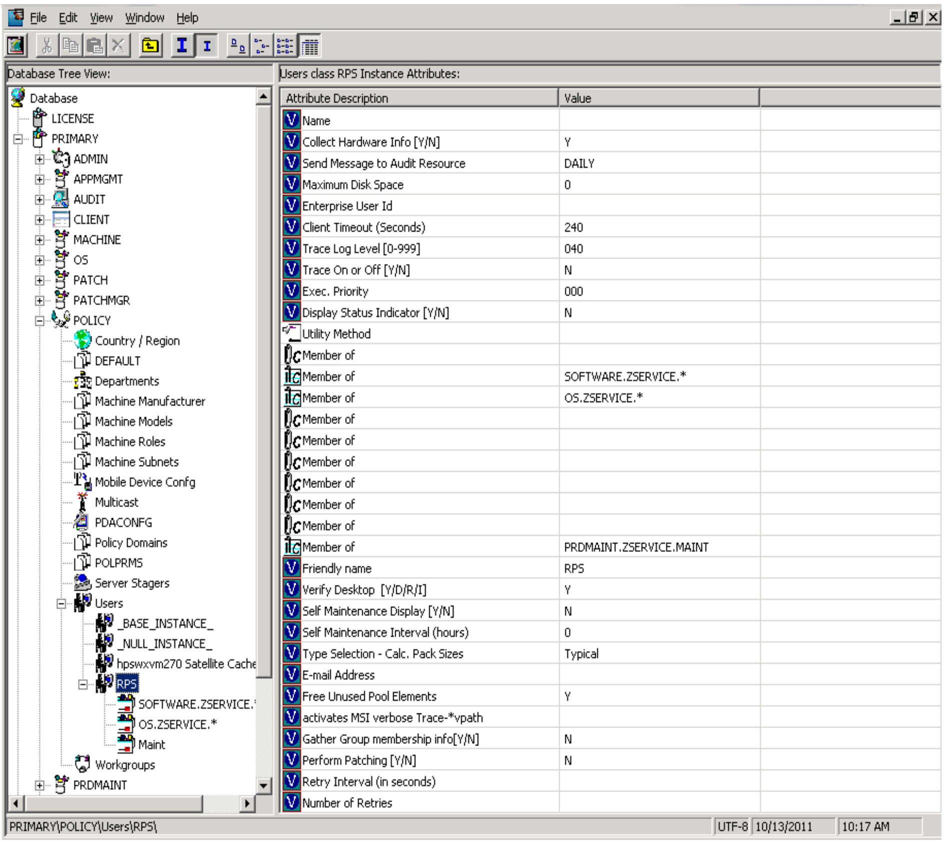
Hardware Auditing

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

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

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

POLICY.USER class – ZCONFIG variable

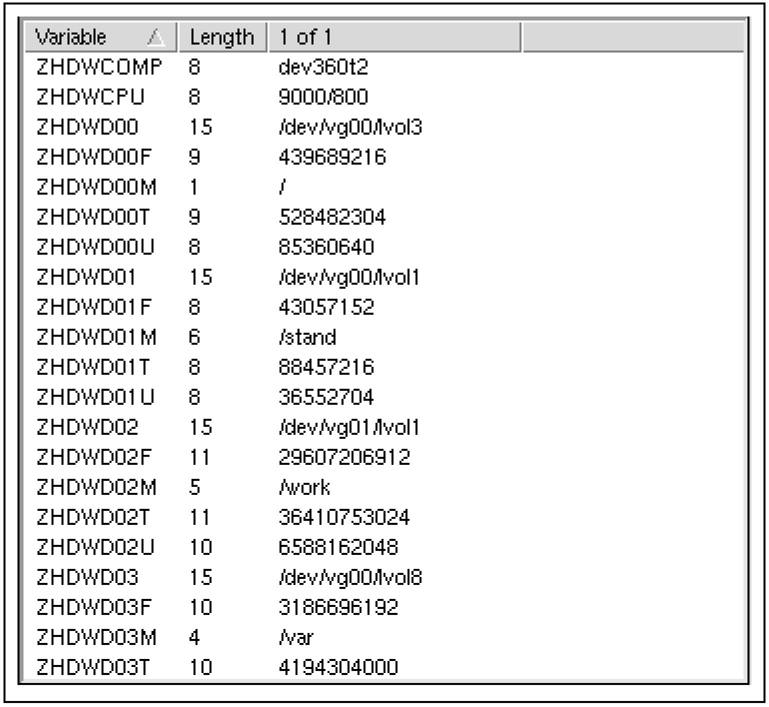


Users class RPS Instance Attributes:

Attribute Description	Value
Name	
Collect Hardware Info [Y/N]	Y
Send Message to Audit Resource	DAILY
Maximum Disk Space	0
Enterprise User Id	
Client Timeout (Seconds)	240
Trace Log Level [0-999]	040
Trace On or Off [Y/N]	N
Exec. Priority	000
Display Status Indicator [Y/N]	N
Utility Method	
Member of	
Member of	SOFTWARE.ZSERVICE.*
Member of	OS.ZSERVICE.*
Member of	
Member of	
Member of	
Member of	
Member of	
Member of	
Member of	PRDMAINT.ZSERVICE.MAINT
Friendly name	RPS
Verify Desktop [Y/D/R/I]	Y
Self Maintenance Display [Y/N]	N
Self Maintenance Interval (hours)	0
Type Selection - Calc. Pack Sizes	Typical
E-mail Address	
Free Unused Pool Elements	Y
activates MSI verbose Trace-*vpath	
Gather Group membership info[Y/N]	N
Perform Patching [Y/N]	N
Retry Interval (in seconds)	
Number of Retries	

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

Sample ZCONFIG object in Admin Agent Explorer



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

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

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

Attributes in a Sample ZCONFIG

Attribute	Description	Example
DESCRIPT	Internal use only	Processing Client Request for &ZCUROBJ
IPADDR01	IP address of network adapter 1	1.1.1.99
LADAPT01	LAN Adapter 1	02608C2CBDCE
LANNUM	LAN Number	1643292
OSREV	Operating System revision number	4
OSVER	Operating System Version	3
ZHDWCPU	CPU Type	000019131C00
ZHDWD00	Drive Name for Drive 00	/dev/hd4
ZHDWD00F	Current free space on drive 00	7028736
ZHDWD00M	Drive 00 mount	/

Attribute	Description	Example
ZHDWD00T	Total space for drive 00	25165824
ZHDWD01	Drive name for drive 01	/dev/hd2
ZHDWD01F	Current free space on drive 01	15859712
ZHDWD01M	Drive 01 mount	/usr
ZHDWD01T	Total space for drive 01	1577058304
ZHDWD02	Drive name for drive 02	/dev/hd9var
ZHDWD02F	Current free space on drive 02	2973696
ZHDWD02M	Drive 02 mount	/var
ZHDWD02T	Total space for drive 02	16777216
ZHDWD03	Drive name for drive 03	/dev/hd3
ZHDWD03F	Current free space on drive 03	28729344
ZHDWD03M	Drive 03 mount	/tmp
ZHDWD03T	Total space on drive 03	41943040
ZHDWDNUM	Number of drive letters assigned	9
ZHDWIPAD	IP address	&(IPADDR01)
ZHDWLANA	LAN Adapter	&(LADAPT01)
ZHDWMEM	Total physical memory (RAM)	65536
ZHDWOS	Operating system and version	Red Hat
ZHDWXHID	Host ID	0x1010163
ZHDWXHN	Host name	linuxdemo
ZOBJRRC	Resolution return code	000
ZOBJRSTY	Resolution type	C
ZSRCCLAS	Service class	ZCONFIG
ZSRCCRC	Service CRC	8B37472C
ZSRCDATE	Service date	20001211
ZSRCDOMN	Service domain	SYSTEMX
ZSRCNAME	Service name	HARDWARE_SCAN
ZSRCPID	Service parent ID	0000000000

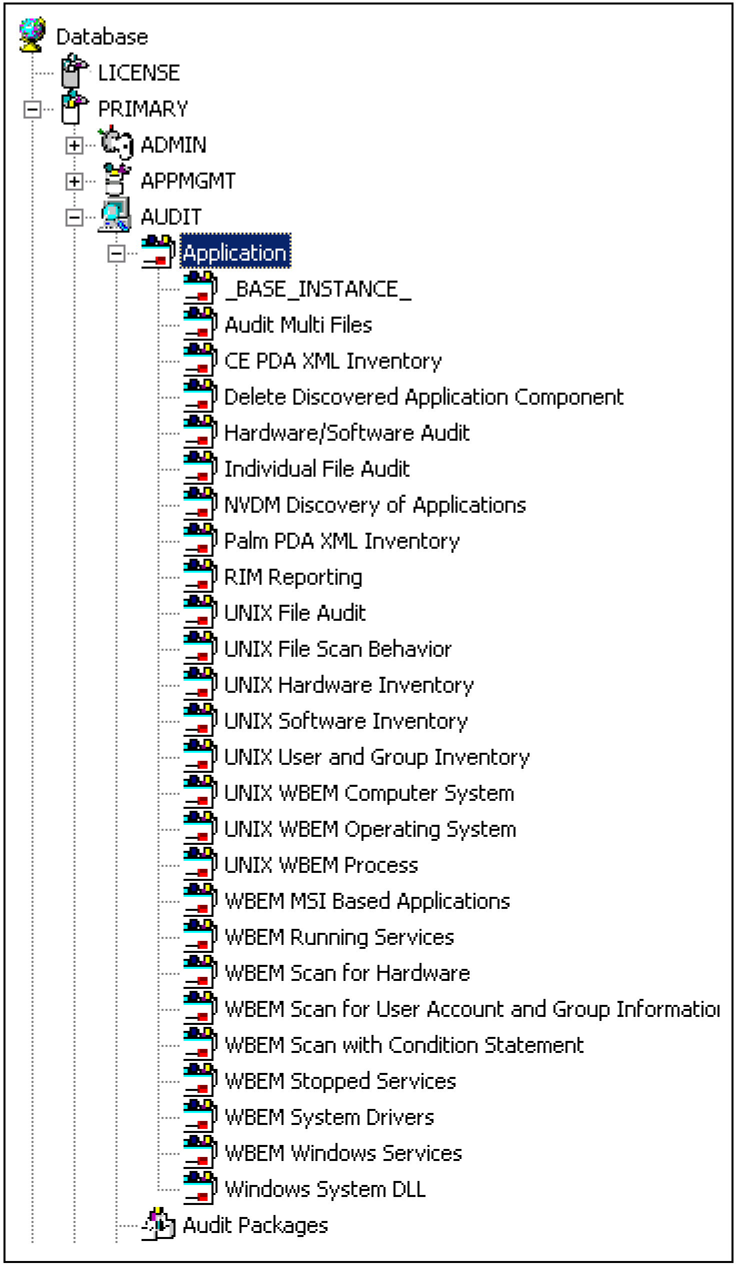
Attribute	Description	Example
ZSRCTIME	Service time	11:52:59
ZUSERID	User ID	jdoe

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

Sample Auditing

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

Sample Auditing Services in PRIMARY.AUDIT Domain



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

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

The sample audit services are described in the following table.

Sample of Auditing Services

Service	Connected to Audit Package (PACKAGE)	Description
BASE		This service instance is the base instance for the Audit

Service	Connected to Audit Package (PACKAGE)	Description
INSTANCE_		Application (ZSERVICE) Class.
Audit Multi Files	Audit to find and Capture Multiple Files	This service scans for a file name or pattern and reports that information back to the administrator.
CE PDA XML Inventory	CE PDA XML Inventory	This service scans for and reports back information on installed Windows CE PDA devices. Will only report back if a device is found.
Delete Discovered Application Component	Audit to Find and Remove Local File	This service looks for a specific file on the user's computer. If it is found, it will be deleted.
Individual File Audit	Audit to Find and Capture Local File	This service performs an NVDM scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes.
NVDM Discovery of Applications	NVDM Discovery of Applications	Used to discover software applications that are installed on an agent machine.
Palm PDA XML Inventory	Palm PDA XML Inventory	This service scans for and reports back information on installed Palm PDA devices. Will only report back if a device is found.
RIM Reporting	RIM Reporting	This service performs a scan of a systems Win32 devices such as: BIOS, Computer System, environment, keyboard, logical disk, logical memory configuration, network adapter, operating system, pointing device, printer, processor product, serial port, service, software element, and video controller. **Note:** This is a very large scan and may take several minutes to complete.
Unix File Scan Audit	UNIX File Scan Audit	This service performs a scan of the user's computer for a specified file of an instance of the AUDIT.FILE classes on UNIX platforms.
Unix Hardware Inventory	Unix Hardware Inventory	This service scans for and reports on a user's hardware on UNIX computers.

Service	Connected to Audit Package (PACKAGE)	Description
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.

WBEMCURR object is not created by default on a Linux 32-bit system when the following Audit services are installed.

- UNIX_WBEM_COMPUTERSYSTEM
- UNIX_WBEM_PROCESS
- UNIX_OPERATINGSYSTEMS

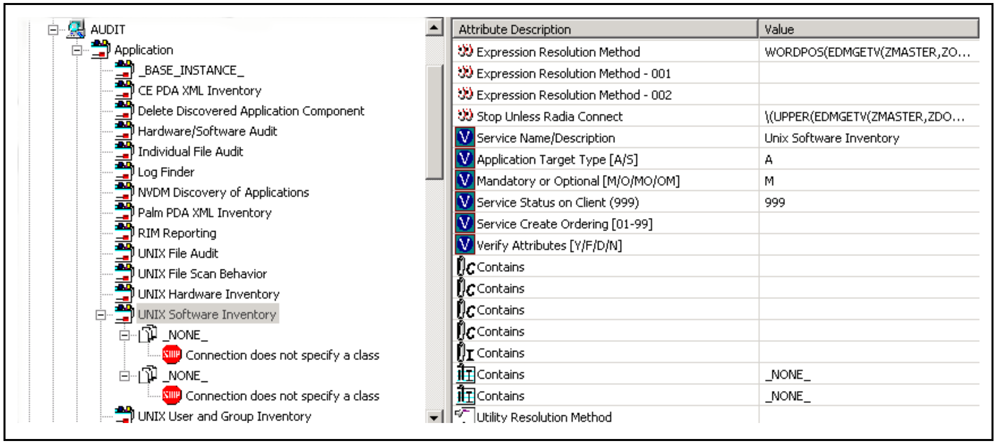
To make Agent Audit services that use WBEM services to work successfully on a Linux 32-bit system, create a softlink libssl.so.0.0.1 and libcrypto.so.0.0.1 with libssl.so and libcrypto.so respectively.Make sure that the CIM Server is started before running the Audit services.

Configuring a Sample Audit

All of the examples presented can be configured for individuals, departments, work-groups, and so forth. For more information on manipulating the database components, see the *Radia Client Automation Enterprise CSDB Editor Online Help*.

For documentation purposes, we will configure the sample audit service Unix Software Inventory. This type of audit scans for all UNIX software that is installed and managed on the agent computer. The ACTION attribute indicates that the discovery of the file will be reported and sent to the Configuration Server for storage.

Unix Software sample audit in AUDIT.ZSERVICE

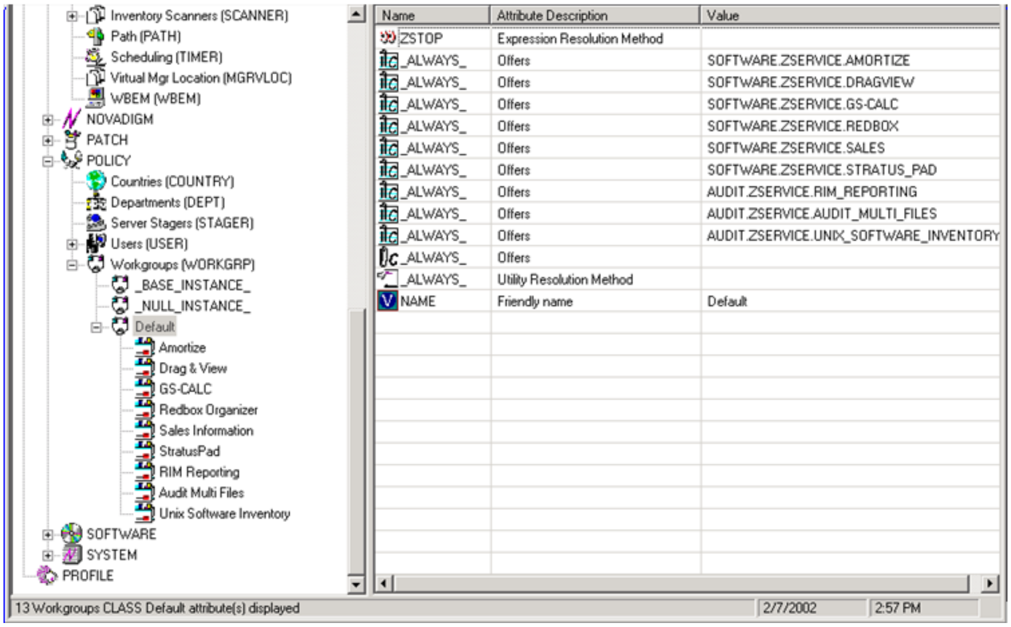


Attribute Description	Value
Expression Resolution Method	WORDPOS(EDMGETV(ZMASTER,ZO...
Expression Resolution Method - 001	
Expression Resolution Method - 002	
Stop Unless Radia Connect	\(UPPER(EDMGETV(ZMASTER,ZDO...
Service Name/Description	Unix Software Inventory
Application Target Type [A/S]	A
Mandatory or Optional [M/O/MO/OM]	M
Service Status on Client (999)	999
Service Create Ordering [01-99]	
Verify Attributes [Y/F/D/N]	
Contains	
Contains	
Contains	
Contains	
Contains	
Contains	_NONE_
Contains	_NONE_
Utility Resolution Method	

To configure a sample Audit package:

1. If you have not already done so, start the Admin CSDB Editor.
2. Navigate to and expand the PRIMARY.AUDIT Domain.
3. Double-click **Application (ZSERVICE)** to expand the class.
4. Scroll to and expand the POLICY Domain.
   For our example, we would like all users who are members of the Workgroup class to select this audit package from their Service Lists.
5. Expand the POLICY.WORKGROUPS class.
6. Select the **Unix Software Inventory** package from the ZSERVICE Class, drag it to the POLICY.WORKGROUPS class, and drop it on the **Default** instance. The Select Connection Attribute window opens.
7. Click **Copy** to add this package.
   The Confirm Connection dialog box opens.
8. Click **Yes** to confirm the connection.

The Unix Software Inventory package is added to WORKGRP Class.



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

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

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

> **Note:** Some scans may take several minutes to complete. This is a normal behavior of the audit scanning process.

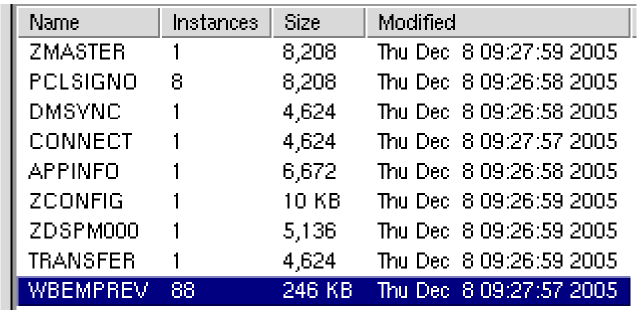
Inventory Scan Results

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

To locate the ZSERVICE object using the Admin Agent Explorer:

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

```
/opt/HP/CM/Agent/lib/SYSTEM/NVDM/SOFTWWARE/ZSERVICE/UNIX_SOFTWARE_
INVENTORY
```



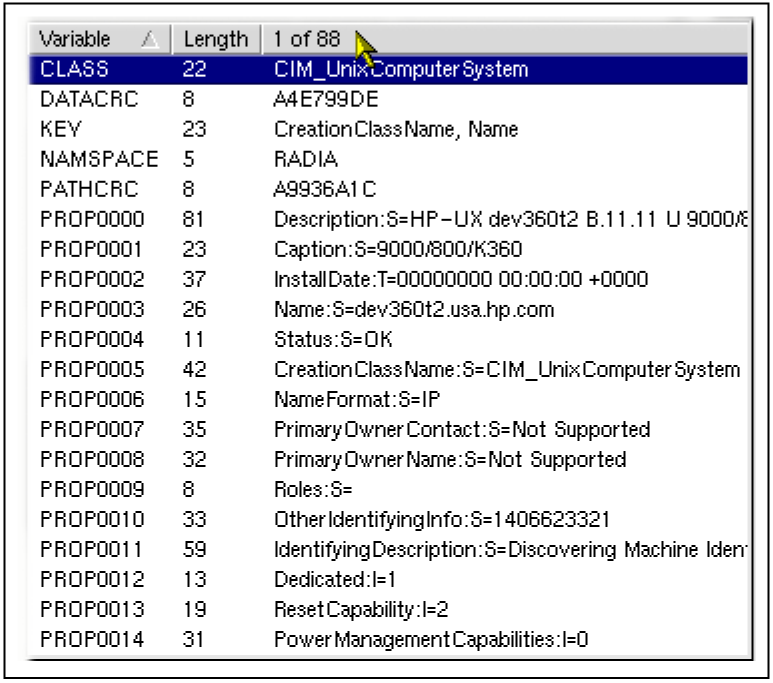
Name	Instances	Size	Modified
ZMASTER	1	8,208	Thu Dec 8 09:27:59 2005
PCLSIGNO	8	8,208	Thu Dec 8 09:26:58 2005
DMSYNC	1	4,624	Thu Dec 8 09:26:58 2005
CONNECT	1	4,624	Thu Dec 8 09:27:57 2005
APPINFO	1	6,672	Thu Dec 8 09:26:58 2005
ZCONFIG	1	10 KB	Thu Dec 8 09:26:59 2005
ZDSPM000	1	5,136	Thu Dec 8 09:26:59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

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

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

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

WBEMPREV heaps in Admin Agent Explorer



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

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

Chapter 4

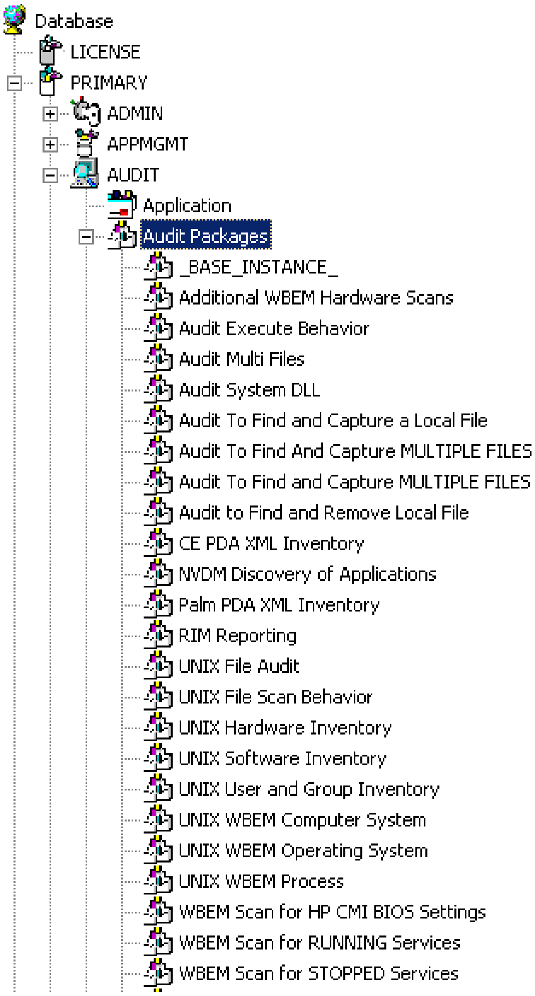
Creating Audit Packages

Audit Packages (PACKAGE) Class

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

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

Audit Packages (PACKAGE) Class

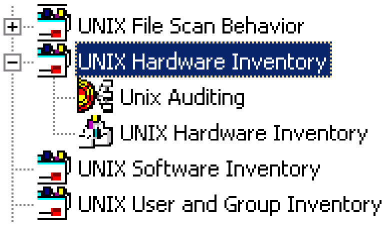


```
Database
    LICENSE
    PRIMARY
        ADMIN
        APPMGMT
        AUDIT
            Application
            Audit Packages
                _BASE_INSTANCE_
                Additional WBEM Hardware Scans
                Audit Execute Behavior
                Audit Multi Files
                Audit System DLL
                Audit To Find and Capture a Local File
                Audit To Find And Capture MULTIPLE FILES
                Audit To Find and Capture MULTIPLE FILES
                Audit to Find and Remove Local File
                CE PDA XML Inventory
                NVDM Discovery of Applications
                Palm PDA XML Inventory
                RIM Reporting
                UNIX File Audit
                UNIX File Scan Behavior
                UNIX Hardware Inventory
                UNIX Software Inventory
                UNIX User and Group Inventory
                UNIX WBEM Computer System
                UNIX WBEM Operating System
                UNIX WBEM Process
                WBEM Scan for HP CMI BIOS Settings
                WBEM Scan for RUNNING Services
                WBEM Scan for STOPPED Services
```

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

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

Unix Hardware Inventory Instance



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

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

> **Note:** 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.

Inventory Options (RIMOPTS) Class

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

Instance	Description
	• ZSVCTYPE attribute is set to I.
Full Audit on Install, Verify and Update	When connected to an audit service, will • Collect attribute is set to Full. • Runexec attribute is set to IVU. • ZSVCTYPE attribute is set to I.

Using Admin CSDB Editor to Create and Maintain Audit Services

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

> **Note:** The Admin CSDB Editor is available for Windows platforms. For more information, see the *Radia Client Automation Enterprise CSDB Editor Online Help*.

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

- What am I auditing for? Will it be a hardware audit, a file audit, or a WBEM object audit?
- Will I be deploying to all users, or a select few?
- Will I want this to be connected to a timer for scheduled deployment?

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

To create a new Audit package:

1. Go to **Start** > **Programs** > **Radia Client Automation Administrator** > **Client Automation Administrator CSDB Editor**
   The Security Information dialog box opens.

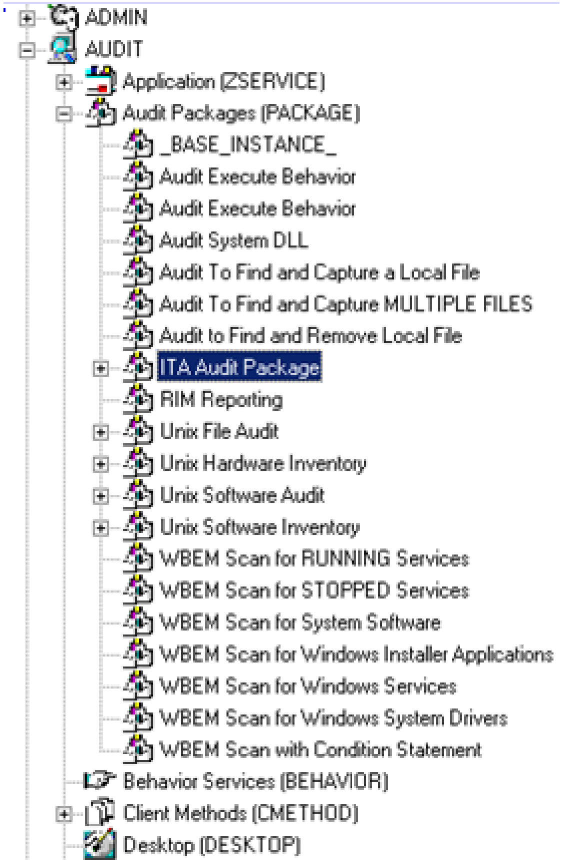
   > **Note:** The default user ID is Admin and password is secret. This might have changed during installation. Check with your RCA security administrator to obtain your own User ID and Password, if necessary.

2. If necessary, type a User ID and Password, and then click **OK**. The Admin CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Expand the **AUDIT** Domain.
5. Double-click the **Audit Packages (PACKAGE) Class**.

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

6. Right-click the **Audit Packages (PACKAGE) Class** and select New Instance from the shortcut menu. The Create Instance dialog box opens.
7. In the upper text box, type a new display name for the package instance. This is the friendly name that will appear in the tree view.
8. In the lower text box, type a name for the Create a new Audit Packages (PACKAGE) instance named. This is the name that appears in the title bar of the list view (right side) of the Admin CSDB Editor window when the instance is selected and opened in the tree view.
9. Click **OK** to continue.

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

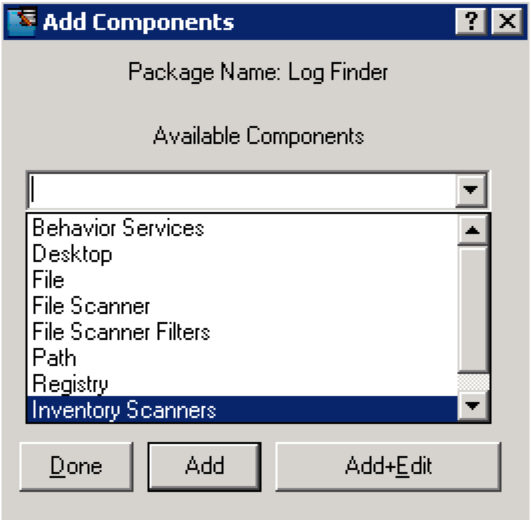

ADMIN
AUDIT
Application (ZSERVICE)
Audit Packages (PACKAGE)
_BASE_INSTANCE_
Audit Execute Behavior
Audit Execute Behavior
Audit System DLL
Audit To Find and Capture a Local File
Audit To Find and Capture MULTIPLE FILES
Audit to Find and Remove Local File
ITA Audit Package
RIM Reporting
Unix File Audit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

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

To add a component to an Audit package:

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

3. Click the **Available Components** down arrow.



4. From the list that opens, select **Inventory Scanners**.
5. In the New Component Name text box, type the name of the new component.
6. Click **Add+Edit**. The component is added to the package and the Editing Instance dialog box opens.
   In the Editing Instance dialog box you can edit the instances that will be used in your audit.
7. Scroll down to the PARMS attribute and select it.
8. In the Parameters text box type `nvdcim`. This is the name of the Tcl script that will be executed by the client to initiate the inventory scan. The nvdcim Tcl script is included with the UNIX Inventory material.

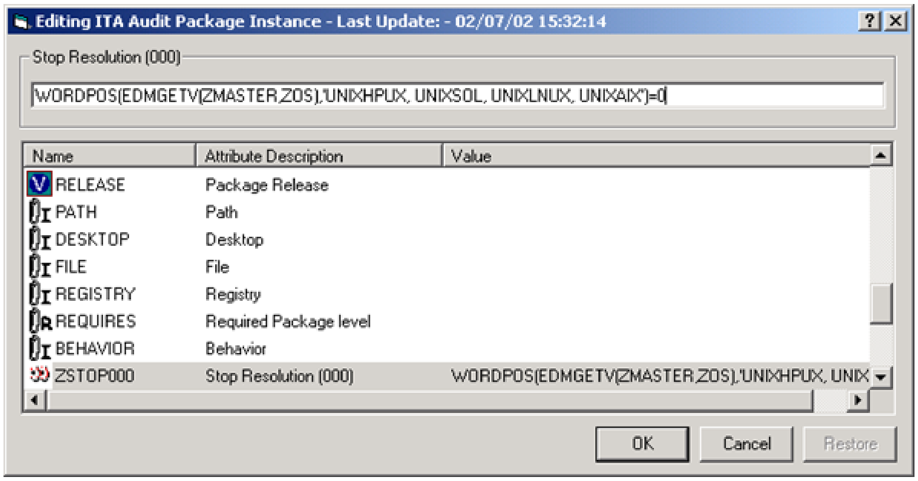
   > **Note:** A connection to the Scanner class may be used to run any custom client inventory method.

9. Click **OK** when you are done with your edit. You return to the Add Components dialog box.
10. Next, add a WBEM Class component to the package. You will need to add a WBEM Class component for each inventory shell script you execute.
11. From the **Available Components** drop-down list, select **WBEM**.
12. In the New Component Name text box, type the name of the WBEM component.
13. Click **Add+Edit**. The component has been added to the package and the Editing CIM_LogicalDisk Instance dialog box opens.
14. Select the **CLASS** attribute, and in the Class text box type `CIM_LogicalDisk`. This is the name of the file that will be used to execute the inventory collection. CLASS is the only attribute used by the client Inventory Harness.
15. When finished, click **OK**.
16. Click **Done** in the Add Components dialog box.

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

To update the ZSTOP expression:

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


Editing ITA Audit Package Instance - Last Update: - 02/07/02 15:32:14
?
OK
Cancel
Restore

Stop Resolution (000)

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

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

4. Click **OK**.

To create a ZSERVICE instance:

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

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

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

   Now, add _NONE_ to the RIMOPTS and BEHAVIOR connections. These are default connections from the base instance and are only applicable to Windows clients.
6. Double click the **ZSERVICE** instance.
7. Double-click the two class connections and change their values to _NONE_.
8. Click **OK**.

Creating UNIX File Audit Methods

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

To create a new Unix File Audit method package:

1. Go to **Start** > **Programs** > **Radia Client Automation Administrator** > **Client Automation Administrator CSDB Editor**
   The Security Information dialog box opens.

   > **Note:** The default user ID is Admin and password is secret. This might have changed during installation. Check with your RCA security administrator to obtain your own User ID and Password, if necessary.

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

To add a component to an audit package:

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

7. Now add a File Scanner Filters component.
8. From the Available Components drop-down list, select **File Scanner Filters**.
9. In the New Component Name text box, type **File Scanner Filters**.
10. Click **Add+Edit** to add the component to the package and open the Editing Instance dialog box.
11. Click **OK** when you are finished editing the instance.
12. Click **Done** in the Add Components dialog box.
13. Now create a ZSERVICE instance and connect the package. Make sure to add _NONE_ to the two ALWAYS connections in the ZSERVICE instance. For instructions on creating a ZSERVICE and removing the required ALWAYS connections, see "To create a ZSERVICE instance:" on page 46 .

Chapter 5

Configuring Timers for Audit Collection

The Scheduling (TIMER) Class

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

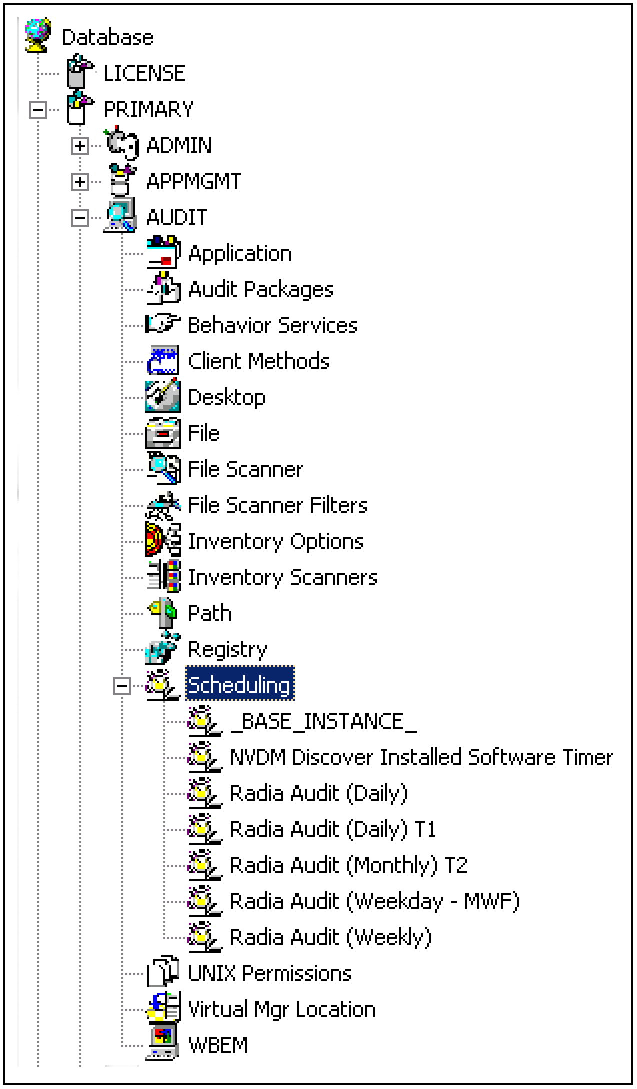
> **Note:** As distributed by HP, the SOFTWARE Domain also contains a Scheduling (TIMER) Class. Timers can be specified in instances of either Scheduling (TIMER) Class and can be connected to an Application (ZSERVICE) Class instance in either the SOFTWARE or AUDIT Domains interchangeably.

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

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

These sample packages can be copied, changing the time parameters to suit your needs. For information on copying an instance, see the *Radia Client Automation Enterprise CSDB Editor Online Help*. Or, you can create a new timer instance by following the procedure "To create a new timer in the AUDIT Domain:" on page 53.

AUDIT Scheduling (TIMER) Class



- Database
  - LICENSE
  - PRIMARY
    - ADMIN
    - APPMGMT
    - AUDIT
      - Application
      - Audit Packages
      - Behavior Services
      - Client Methods
      - Desktop
      - File
      - File Scanner
      - File Scanner Filters
      - Inventory Options
      - Inventory Scanners
      - Path
      - Registry
      - Scheduling
        - _BASE_INSTANCE_
        - NVDM Discover Installed Software Timer
        - Radia Audit (Daily)
        - Radia Audit (Daily) T1
        - Radia Audit (Monthly) T2
        - Radia Audit (Weekday - MWF)
        - Radia Audit (Weekly)
      - UNIX Permissions
      - Virtual Mgr Location
      - WBEM

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

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

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

Scheduling (TIMER) Class

Attribute	Description
ZOBJPRI	Sets the priority for deployment of ZTIMEQ object, which is deployed relative to the other elements being deployed during the agent connect. Elements with a priority number less than the value of ZOBJPRI are deployed before the ZTIMEQ

Attribute	Description
	object. A value of 90 is inherited from the base instance and should not be changed.
ZSTOP	Used to assign timer conditions. Indicate true to cause resolution of the instance to be skipped. The timer is not deployed for end users. Leave blank for the instance to be accepted, and resolution will continue.
ZSCHMODE	Specifies the timer owner. It is recommended that you accept the default configuration of USER.
ZSCHDEF	Indicates when timer expires. The syntax varies depending on the frequency of expiration, which can be DAILY, HOURLY, INTERVAL, NUMDAY, WEEKDAY, WEEKLY.
ZSCHTYPE	Used *only when* ZSCHFREQ = PERIODIC. Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4 AM (non-peak) scheduled event that is sent to the agent computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning. Example 1: Suppose you create and deploy a timer with the ZSCHDEF = DAILY(&ZSYSDATE,04:00:00) If ZSCHTYPE = IMMEDIATE and it is: • Before 04:00:00, the command in the instance will be executed the same day at 04:00:00 • After 04:00:00, the command in the instance will be executed immediately If ZSCHTYPE = DEFERRED and it is: • Before 04:00:00, the command in the instance will be executed the next day at 04:00:00 • After 04:00:00, the command in the instance will be executed the next day at 04:00:00 Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,04:00:00) If ZSCHTYPE = IMMEDIATE and it is: • Not Friday or Friday and before 04:00:00, the command in the instance will be executed on Friday at 04:00:00 • Friday and after 04:00:00, the command in the instance will be executed immediately

Attribute	Description
	If ZSCHTYPE = DEFERRED and it is: • Not Friday or Friday and before 04:00:00, the command in the instance will be executed a week later on Friday at 04:00:00 • Friday and after 04:00:00, the command in the instance will be executed a week later on Friday at 04:00:00
ZSCHFREQ	Indicates how often the timer should expire according to the frequency specified in the ZSCHDEF attribute. • Once for a one-time expiration. • Periodic for a repeated expiration. • Random for random intervals.
ZRSCCMDL	Indicates the command line that is executed on the subscriber's computer when the timer expires.
ZSVCOID	Specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified.
ALWAYS	Stores the connections to other instances.
NAME	The friendly name for this instance.
APPSVC	The Application Name.
REQUEST	The Application Request.
DOMAIN	The server's domain name.
IPADDR	The server's IP address/name.
SOCKET	The server's socket number.
MGRNAME	The server's name.
ZCREATE	The Scheduler CREATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZVERIFY	The Scheduler VERIFY method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZUPDATE	The Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZDELETE	The Scheduler DELETE method that runs on the agent computer. This value is inherited from the base instance and should not be changed.
ZNOPING	Controls the automatic sensing of a network connection between the agent

Attribute	Description
	computer and the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. Use this attribute when there is a possibility that the client will not be able to connect with the Configuration Server, such as when the client is a mobile user. **Note:** In order to use this attribute, you must add it to the TIMER Class template.

Creating a Timer Instance

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

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

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

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

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

For more information concerning the Scheduling (TIMER) Class, see the *Radia Client Automation Enterprise Administrator User Guide*.

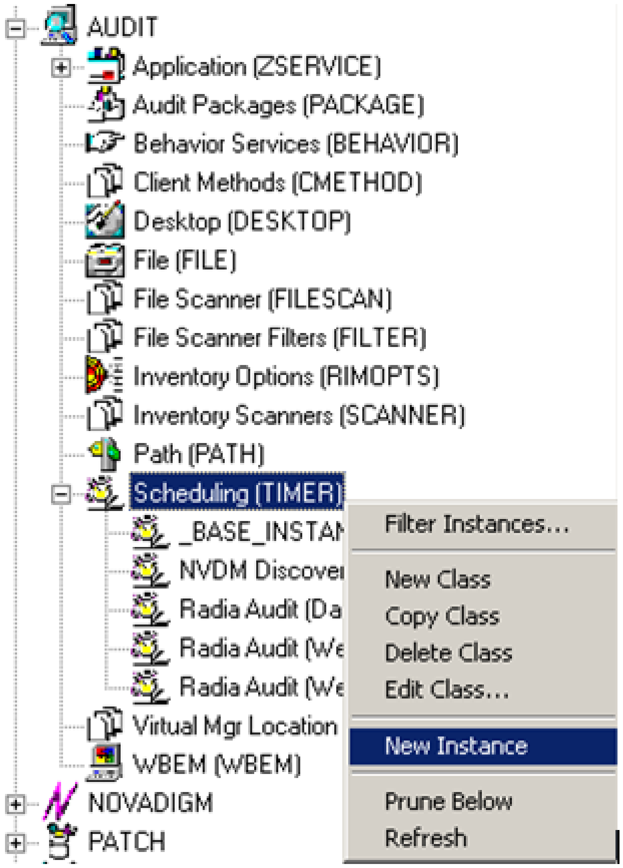
> **Note:** The following section uses the Admin CSDB Editor, which is available for 32-bit Windows platforms.

To create a new timer in the AUDIT Domain:

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

> **Note:** The default user ID is Admin and password is secret. This might have changed during installation. Check with your RCA security administrator to obtain your own User ID and Password, if necessary.

2. If necessary, type a User ID and Password, and then click **OK**. The Admin CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Expand the **AUDIT Domain**.
5. Right-click **Scheduling (TIMER)**.
   A shortcut menu opens.



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

Specifying Timer Settings

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

> **Note:** For more information concerning the Scheduling (TIMER) Class, see the *Radia Client Automation Enterprise Administrator User Guide*.

Specifying ZSCHDEF

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

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

- The value of *freq* can be:
  `DAILY, WEEKLY, WEEKDAY, HOURLY, INTERVAL, NUMDAYS`
  - If the value of *freq* is `DAILY, WEEKLY, HOURLY, INTERVAL, or NUMDAYS`, the date is then specified in the following form:
    `YYYY/MM/DD`
  - If the value of *freq* is `WEEKDAY`, the `date` is then specified as the name of a day of the week in all uppercase letters. This would be one of the following:
    `MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY`
- The value for `time` or `limit_time` is optional. It is specified in the following form:
  `HH:MM:SS`
- The value for `count` is optional. It is specified as an integer.
- The timer expiration can also be configured on the value of ZSCHFREQ. Use the following table to help you determine the appropriate syntax.

Syntax of ZSCHDEF Variables

Type	Syntax	Timer Expires
DAILY	DAILY(&ZSYSDATE,24:00:00)	Daily at midnight by the system's date.
WEEKLY	WEEKLY(&ZSYSDATE,01:00:00)	Every 7 days at 1:00 AM.
WEEKDAY	WEEKDAY(MONDAY,01:00:00)	Every Name of Weekday* starting on MONDAY at 1:00 AM. The weekday must be specified in uppercase.
HOURLY	HOURLY(&ZSYSDATE,08:41:00)	Hourly starting at 8:41 AM on the systems date.
INTERVAL	INTERVAL(&ZSYSDATE,08:41:00,,30)	Every 30 minutes starting at 8:41 AM based on system's date.
NUMDAYS	NUMDAYS(20000803,08:00:00,,14)	Every 14 days starting on August 3, 2000 at 8:00 AM.

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

Specifying ZSCHTYPE

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

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

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

Specifying ZSCHFREQ

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

Specifying ZRSCCMDL

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

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

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

> **Note:** The parameters indicated in the radskman command may differ depending upon customer specific implementations.

Specifying ZNOPING

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

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

Connecting the Timer to a Service

Once you have created your timer, you must connect it to a service. Each subscriber that receives the ZSERVICE to which the timer is connected, will receive the timer information in the ZTIMEQ object the next time the agent connects to the Configuration Server.

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

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

Audit Execution Configuration

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

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

- Customizing the Inventory Options (RIMOPTS) attribute.

and

- Updating the embed tags in the html file for the Application Self-service Manager.

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

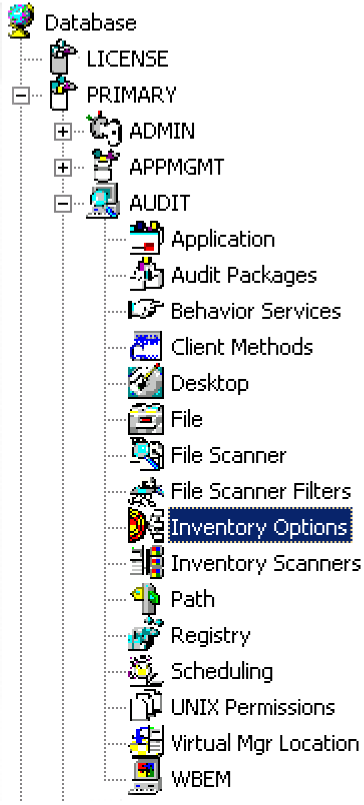
To customize the RIMOPTS instance:

1. Go to **Start** > **Programs** > **Radia Client Automation Administrator** > **Client Automation Administrator CSDB Editor**
   The Security Information dialog box opens.

   Note: The default user ID is Admin and password is secret. This might have changed during installation. Check with your RCA security administrator to obtain your own User ID and Password, if necessary.

2. If necessary, type a User ID and Password, and then click **OK**. The System Explorer window opens.

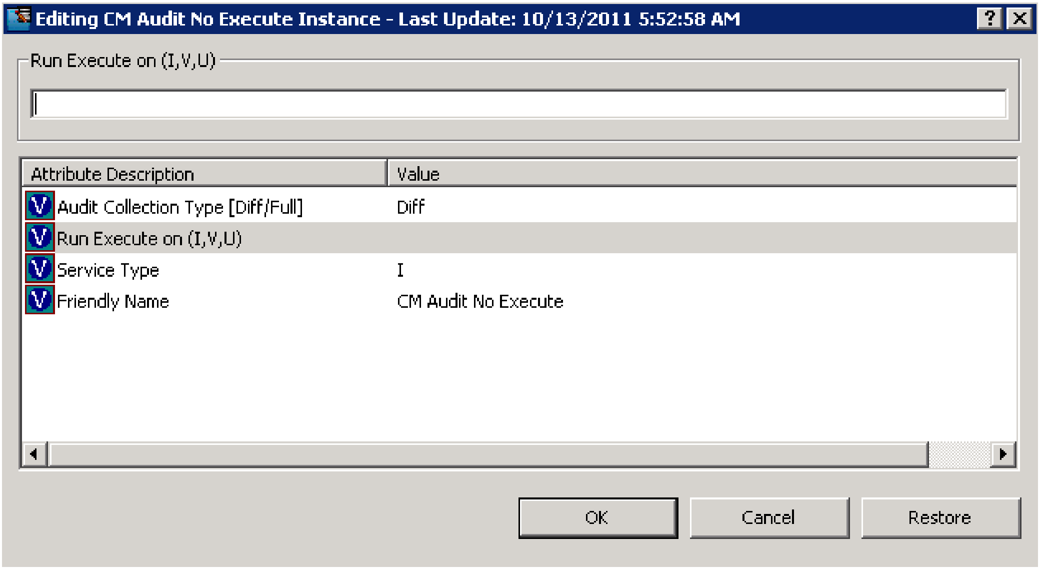
3. Expand the **PRIMARY File** and the **AUDIT Domain**.



4. Create a new instance in the Inventory Options (RIMOPTS) Class called CM_AUDIT_NO_EXECUTE, and click **OK**.

5. Expand the Inventory Options (RIMOPTS) Class and double-click the **CM Audit No Execute** instance.

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


Editing CM Audit No Execute Instance - Last Update: 10/13/2011 5:52:58 AM

Run Execute on (I,V,U)

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

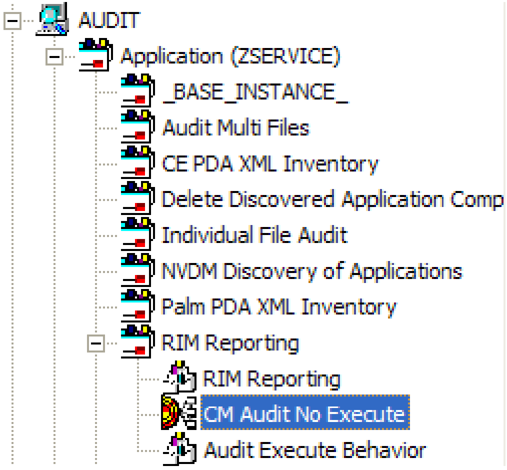
Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the RIM_REPORTING service.

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

8. Select **Edit Instance**.

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

10. Next, to define the audit service as Mandatory, locate the **ZSVCMO** field and set it to M. This will cause the initial TIMER definition associated with the audit service to be created on the agent.
The **CM Audit No Execute** instance is now connected to the RIM Reporting service, as shown in the following figure:



Chapter 6

Viewing Inventory Reports

To view the inventory reports:

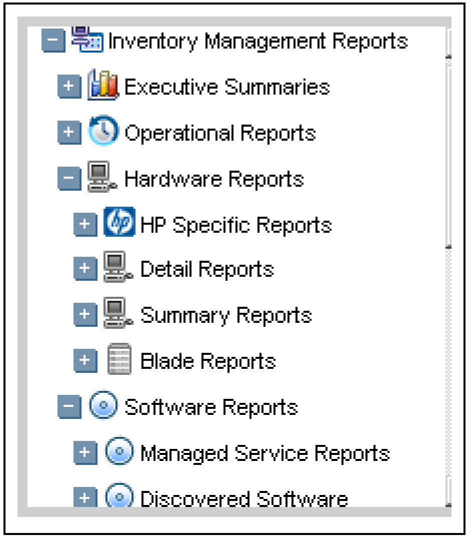
1. Go to your RCA Console.
2. Click the Reporting tab.
3. Select the Inventory Management Reports under Reporting Views.

Reporting Views for Inventory Reports

There are different types of inventory reports:

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

Inventory Management Reports



Inventory Management Reports
Executive Summaries
Operational Reports
Hardware Reports
HP Specific Reports
Detail Reports
Summary Reports
Blade Reports
Software Reports
Managed Service Reports
Discovered Software

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

Hardware Reporting Views

Reporting View Types	Reporting Views
HP Specific Reports	HP BIOS Settings

Reporting View Types	Reporting Views
	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 TPM Chipset Information S.M.A.R.T . Alerts
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

Software Reporting Views

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

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

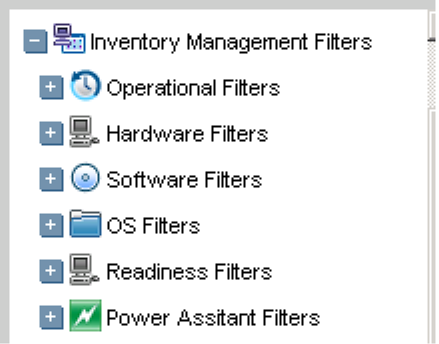
Filtering Inventory Reports with Reporting Server

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

Filter types include:

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

Inventory Management Related Data Filters



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

For more information on creating filters and using the Reporting tab in general, see the *Radia Client Automation Enterprise User Guide*.

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click here.

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to radiadocfeedback@persistent.co.in.

Product name and version: Radia Client Automation Enterprise Inventory Manager, 9.00

Document title: Reference Guide

Feedback: