

HP Client Automation Enterprise

Inventory Manager

for the Windows® operating system

Software Version: 7.20

Installation and Configuration Guide

Manufacturing Part Number: None

Document Release Date: August 2008

Software Release Date: July 2008



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

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

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

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

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

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

TFTP SERVER

Copyright © 1983, 1993

The Regents of the University of California.

OpenLDAP

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

OpenSSL License

Copyright © 1998-2001 The OpenSSLProject.

Original SSLeay License

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

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
 - The number before the period identifies the major release number.
 - The first number after the period identifies the minor release number.
 - The second number after the period represents the minor-minor release number.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

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

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

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

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

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

Table 1 lists new features added for this release of Configuration Management 7.20.

Table 2 lists features added for the Configuration Management v 5.xx releases.

Table 1 New features for this Release

| Chapter | Version | Changes |
|---------|------------------|---|
| All | 7.20 | Most components in the HP Configuration Management Version 5.1x product family have been rebranded to HP Client Automation for Version 7.20. |
| 1 | 7.20 | Page 14, Using this Guide with Core and Satellite Servers , new topic. |
| 2 | 7.20 Aug 2008 | Page 27, Creating the Database and DSN for Inventory Manager , modified topic title, and added Roles and Privileges to the defined user profile needed to create the database using Oracle. |
| 3 | 7.20 | Page 31, Installing the Client Automation Agents , the Application Manager Agent includes embedded support for the components supported by the earlier Server Management Agent; |

| Chapter | Version | Changes |
|---------|---------|---|
| | | Application Management Profiles and Windows Terminal Services and Citrix support. The Server Management Agent feature has been removed from the Client Automation Agent installation. |
| 4 | All | Page 71, Implementing Registry Scans , topic modified to use the Add Component dialog to both Add and Edit a registry scan instance from an AUDIT.PACKAGE instance. |
| 4 | 7.20 | Removed the topic Updating the CSDB for Registry Scans; the PRIMARY.AUDIT.REGISTRY class is included in the default CSDB. |
| 4 | All | Page 68, RIMOPTS Class , corrected the procedure: To enable drag-and-drop connections for RIMOPTS Class instances . Changed the CSDB class needed to access the CONNECT_ZSERVICE_TO_RULES instance from: PRIMARY → ADMIN → Name Lists (32) (ZLIST32) to PRIMARY → ADMIN → Name Lists (8) (ZLIST) . |
| 8 | all | Page 124, Table 20 , Scheduling Timer Class, ZSCHTYPE Row, timer examples now show the required leading zero to indicate 4 AM: e.g., ZSCHDEF = DAILY(&ZSYSDATE,04:00:00). |

Table 2 New features added for Configuration Management 5.xx

| Chapter | Version | Changes |
|---------|---------|--|
| | 5.10 | The Administrator installation chapter was removed and is now included in the Administrator Guide. |
| 3 | 5.00 | Page 25, Creating the Inventory Manager Environment , new chapter explains how to define an ODBC database and DSN for Inventory data, and how to setup the various Infrastructure servers used to collect (Configuration Server), post (Messaging Server), and report (Reporting Server) the data. |
| 3 | 5.00 | Page 32, System Requirements for installing the agents have changed. |

| Chapter | Version | Changes |
|---------|---------|--|
| 5 | 5.00 | Page 87, WBEM Object Processing , discusses how WBEM objects are collected and posted by the WBEM Data Delivery Agent of the Messaging Server to the Inventory ODBC database. Replaces the previous topic, WBEM Objects and the Configuration Server. |
| 9 | 5.00 | Page 139, Viewing Inventory from the Reporting Server , new chapter. Replaces the earlier chapter: “Viewing Inventory from the Radia Integration Server”. |
| 10 | 5.00 | Page 143, Windows Vista Readiness Reports , new reports are available from the Inventory Reports within the Reporting Server. |

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 13 |
| | About the Inventory Manager | 14 |
| | Using this Guide with Core and Satellite Servers | 14 |
| | Overview..... | 15 |
| | About the Reporting Server | 16 |
| | Inventory Manager Terminology | 18 |
| | Client Automation Prerequisites..... | 19 |
| | Necessary Skills..... | 20 |
| | Inventory Manager Technology | 20 |
| | Common Information Model (CIM)..... | 20 |
| | Web-Based Enterprise Management (WBEM) | 21 |
| | Windows Management Instrumentation (WMI)..... | 21 |
| | Client Automation and WBEM..... | 22 |
| | About this Guide..... | 22 |
| 2 | Creating the Inventory Manager Environment..... | 25 |
| | Inventory Manager Implementation Tasks | 26 |
| | Configuration Server and Database | 26 |
| | SQL or Oracle Prerequisites..... | 27 |
| | Creating the Database and DSN for Inventory Manager | 27 |
| | Administrator CSDB Editor | 29 |
| | Messaging Server | 29 |
| | Reporting Server | 30 |
| | Portal (Optional) | 30 |
| 3 | Installing the Client Automation Agents..... | 31 |
| | Client Automation Agent Installation..... | 32 |

| | |
|--|-----------|
| System Requirements | 32 |
| Platform Support..... | 32 |
| Client Automation Agent Installation Process | 33 |
| Preparing Install.ini | 34 |
| [PROPERTIES] Section of Install.ini..... | 36 |
| [ARGS] Section of Install.ini..... | 40 |
| [OBJECTS] Section of Install.ini..... | 44 |
| Installing the Client Automation Agents..... | 45 |
| Installing the Client Automation Agent from a Command Line | 46 |
| Referencing the Client Automation Agents | 46 |
| Specifying the Client Automation Agent Features to Install | 47 |
| Additional Command Line Arguments | 47 |
| Examples..... | 48 |
| Initiating the Client Automation Agent Installation from a Logon Script | 49 |
| Using the Agent Installation Wizard..... | 50 |
| Removing the Client Automation Agents | 52 |
| Using the Installation Wizard to Remove Client Automation Agents | 52 |
| Using a Command Line to Remove Client Automation Agents..... | 53 |
| Repairing the Client Automation Agents..... | 54 |
| Using the Installation Wizard to Repair Client Automation Agents | 54 |
| Using a Command Line to Repair Client Automation Agents..... | 55 |
| Modifying the Client Automation Agent Installation | 55 |
| Using the Installation Wizard to Modify the Client Automation Agent Installation.. | 55 |
| Using a Command Line to Modify the Client Automation Agent Installation | 56 |
| Using a Pre-Install Script | 57 |
| Using a Post-Install Script..... | 58 |
| Local AIP Support for the MSI Redirector..... | 59 |
| Internet Proxies | 60 |
| 4 The AUDIT Domain | 63 |
| The AUDIT Domain in the CSDB | 64 |
| AUDIT Domain Defined | 64 |
| RIMOPTS Class | 67 |
| REGISTRY Class..... | 69 |

| | |
|---|------------|
| Implementing Registry Scans | 71 |
| Inventory Database Tables | 72 |
| 5 Software and Hardware Auditing | 75 |
| Auditing Types | 76 |
| File Auditing..... | 76 |
| WBEM Auditing..... | 83 |
| WBEM Object Processing | 87 |
| Disabling Remnant Configuration Server Instances for WBEM Object Processing | 87 |
| Hardware Auditing | 88 |
| 6 Successful Auditing | 93 |
| Sample Auditing | 94 |
| Configuring a Sample Audit | 97 |
| What Happened during the Scan? | 101 |
| 7 Creating Audit Packages | 105 |
| Audit Packages or PACKAGE Class | 106 |
| Using the CSDB Editor Create/Maintain Audit Services | 109 |
| 8 Configuring Timers for Audit Collection | 121 |
| The Scheduling (TIMER) Class | 122 |
| Creating a Timer Instance | 127 |
| Specifying Timer Settings | 129 |
| Specifying ZSCHDEF..... | 129 |
| Specifying ZSCHTYPE | 131 |
| Specifying ZSCHFREQ..... | 131 |
| Specifying ZRSCCMDL..... | 131 |
| Specifying ZNOPING, PINGDLAY, and PINGCNT | 132 |
| Connecting the Timer to a Service | 133 |
| Audit Execution Configuration..... | 134 |

| | |
|--|-----|
| 9 Viewing Inventory from the Reporting Server | 139 |
| Accessing the Reporting Server | 140 |
| Viewing Audit Information Using the Reporting Server | 140 |
| Reporting Views for Inventory Reports | 140 |
| Windows Vista Readiness Reports..... | 143 |
| Filtering Inventory Reports with Reporting Server | 143 |
| A Detail and Summary Reporting Tables | 147 |
| Index | 159 |

1 Introduction

At the end of this chapter, you will:

- Understand the components of the HP Client Automation Inventory Manager (Inventory Manager) agent.
- Understand the terminology associated with the Inventory Manager agent.
- Be familiar with supported platforms.
- Be familiar with WBEM and the Client Automation agent.
- Be familiar with related Client Automation infrastructure components for processing inventory agent data, such as the HP Client Automation Messaging Server (Messaging Server), the HP Client Automation Reporting Server (Reporting Server), and the HP Client Automation Portal (Portal).

About the Inventory Manager

The Inventory Manager is an agent utility used with the auditing software available in the Audit Domain of the Configuration Server Database (CSDB) to discover configuration information on remote computers. It enables centralized reporting and administration based upon the discovery results.

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

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

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

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

Using this Guide with Core and Satellite Servers



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

Overview


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

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

- **Application Self-service Manager** installs the software when the user selects an application to be installed.
- **Application Manager** installs the software in one of the following ways:
 - through a logon script.
 - when the user double-clicks a desktop HP Connect icon.
 - according to a schedule.
 - using the Notify capabilities of the Portal or the Configuration Server.

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

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

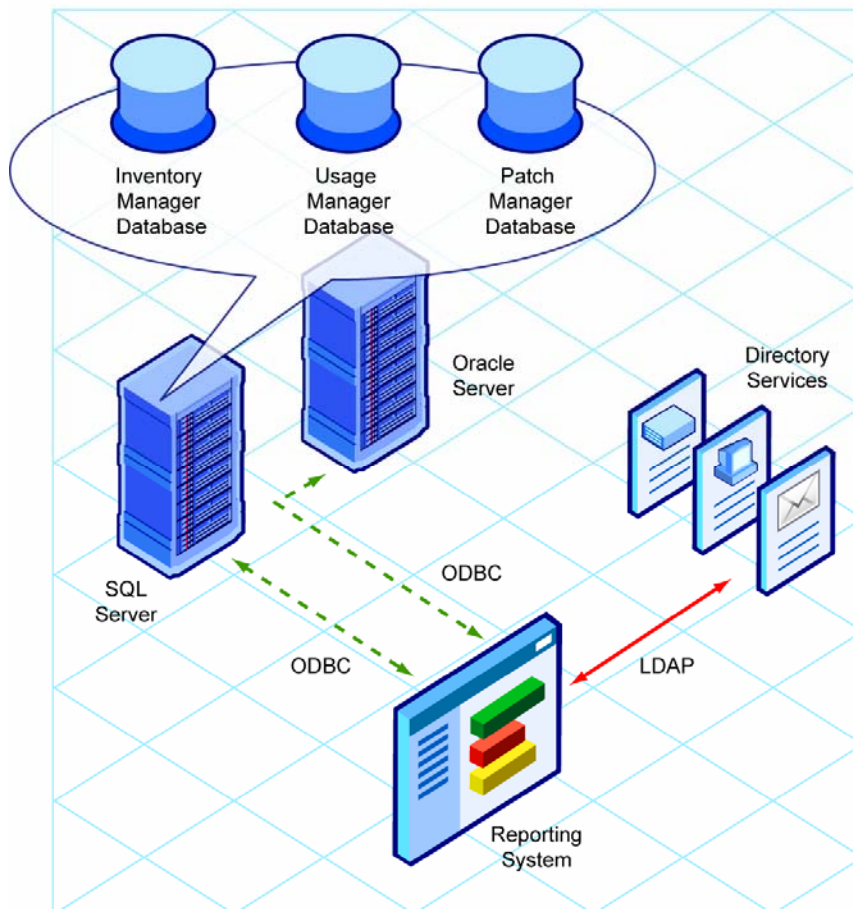
 By default, the Data Directory Agents of the Messaging Server post the inspection results directly to the ODBC database.

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

About the Reporting Server

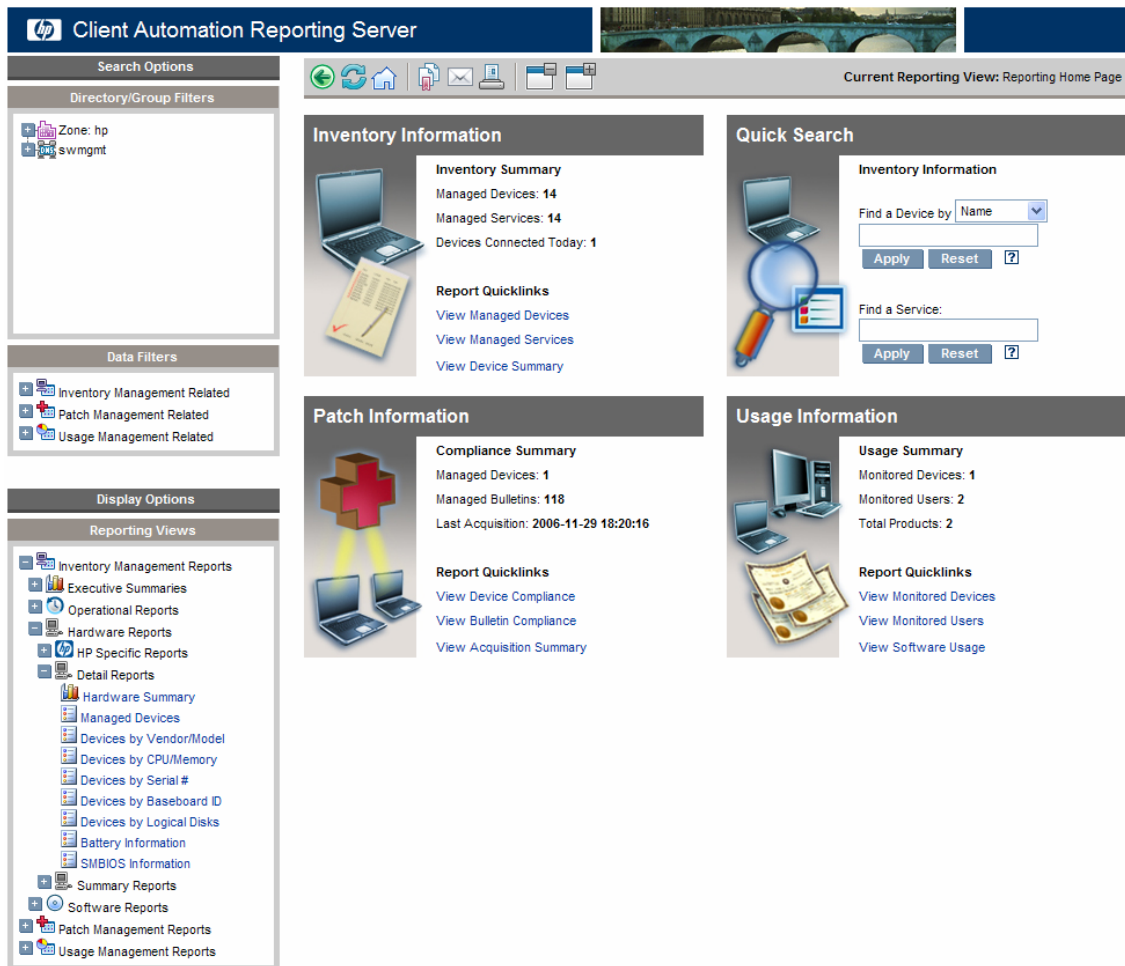
As part of the extended infrastructure for Client Automation, the web-based Reporting Server allows you to query the combined data in existing Inventory Manager, Patch Manager, and Application Usage Manager databases and obtain Executive, Summary, and Detailed reports. In addition, you have the option of mounting an existing LDAP directory, which allows you to filter your data using your LDAP directory levels. The Reporting Server environment is illustrated in the figure below.

Figure 1 Reporting environment



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

Figure 2 Reporting Server Web interface supports auditing



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

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

Inventory Manager Terminology

agent computer

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

CIM (Common Information Model)

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

clean machine

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

Client Automation agent

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

Messaging Server

The Messaging Server is the Client Automation infrastructure component that provides a common routing and inter-server data delivery service, especially for report-bound data. When servicing a Configuration Server, the Messaging Server handles the delivery of Inventory, Patch, and Portal data collected from Client Automation agents to the appropriate external location. Data Delivery Agents are used to post data directly to an SQL-compliant database using ODBC.

Portal

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

Reporting Server

The Reporting Server is a Web-based interface to the reportable data captured by the Client Automation extended infrastructure product suite. It

allows you to query the combined data in existing Inventory Manager, Patch Manager, and Application Usage Manager databases and create detailed reports. You have the option of mounting an existing LDAP directory, which allows you to filter your data using your LDAP directory levels.

Web-Based Enterprise Management (WEBM)

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

Windows Management Instrumentation (WMI)

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

WMI Repository

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

Client Automation Prerequisites

The Inventory Manager requires the following Client Automation components:

- Configuration Server
- Client Automation agent
 - Application Manager
- and/or
 - Application Self-service Manager
- Messaging Server. Refer to the *Messaging Server Guide* for more information on installing or migrating to the Messaging Server, and how the Messaging Server transfers data directly, or indirectly, to an Inventory Manager database.

Necessary Skills

Client Automation Products

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

Web-Based Enterprise Management

This document assumes that the reader is familiar with Web-Based Enterprise Management (WBEM). Resources for familiarizing yourself with WBEM can be found at the following web site:

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

Microsoft Implementations of WBEM

This document also assumes that the reader is familiar with Windows Management Instrumentation (WMI). Information concerning WMI can be found at the following web site:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/w98ddk/hh/w98ddk/wmi_wp_03se.asp.

Inventory Manager Technology

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

Common Information Model (CIM)

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

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

- Allowing for the definition of **associations**, such as describing relationships between object-dependencies, component relationships, and connections.
- Allowing for the definition of **methods**, such as input/output parameters and return codes.

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

Web-Based Enterprise Management (WBEM)

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

Windows Management Instrumentation (WMI)

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

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

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



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

Client Automation and WBEM

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



Windows 2000 and above systems have WMI natively installed. If necessary, WMI version 1.5 is available for download by licensed end users from the Microsoft download center at

<http://www.microsoft.com/downloads/details.aspx?FamilyID=afe41f46-e213-4cbf-9c5b-fbf236e0e875&DisplayLang=en>.

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

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



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

About this Guide

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

- **Creating the Inventory Manager Environment**
This chapter describes how to define an ODBC database and DSN, and setup the related Client Automation infrastructure servers needed to collect, post, and report the inventory audit information.
- **Installing the Client Automation agents**
This chapter describes how to install the Client Automation Inventory Agents.
- **The AUDIT Domain**
This chapter discusses the AUDIT Domain within the CSDB.
- **Software and Hardware Auditing**
This chapter discusses the different types of auditing, how to configure a supplied audit service, and how to create your own audit service.
- **Configuring Timers for Audit Collection**
This chapter discusses how to perform audits using a timer.
- **Viewing Inventory from the Reporting Server**
This chapter discusses the topics related to viewing inventory data and reports from the Reporting Server.

2 Creating the Inventory Manager Environment

At the end of this chapter, you will:

- Be familiar with the tasks needed to set up the Client Automation infrastructure and SQL or Oracle database for an Inventory Manager environment.
- Create a SQL or Oracle database for the Inventory Manager data and an ODBC DSN connection to it.
- Have the following infrastructure installed and configured to support an Inventory Manager environment:
 - HP Client Automation Configuration Server (Configuration Server) and HP Client Automation Configuration Server Database (CSDB)
 - HP Client Automation Administrator Configuration Server Database Editor (Admin CSDB Editor)
 - HP Client Automation Messaging Server (Messaging Server)
 - HP Client Automation Reporting Server (Reporting Server)
 - HP Client Automation Portal (Portal) (optional)

Inventory Manager Implementation Tasks



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

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

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

Configuration Server and Database

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

SQL or Oracle Prerequisites

Before setting up your environment for the Inventory Manager, you must have already installed the latest version of Microsoft SQL Server 2000 Service Pack 3a or greater. Microsoft SQL Server 2005 is also supported.

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

Creating the Database and DSN for Inventory Manager

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



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

To create the Inventory database using Microsoft SQL Server

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

| | |
|---------------------|--|
| General tab | Name: CMAUDIT (or name of your choice with no blanks or underscores) |
| Data Files tab | Initial Size: 500 MB Select Autogrow by 20%. |
| Transaction Log tab | Change initial size: 100 MB |

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

To create the Inventory database using Oracle

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

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

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

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

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

| | |
|----------|---|
| Username | cminventory |
| Password | Create one based on your enterprise's security recommendations. |

| | |
|----------------------|---|
| Default tablespace | INVENTORYDATA |
| Temporary tablespace | INVENTORYTEMP |
| Profile | DEFAULT or a PROFILE NAME used for this schema) |
| Roles | CONNECT and RESOURCE |
| System Privileges | CREATE ANY VIEW SELECT ANY TABLE UNLIMITED TABLESPACE UPDATE ANY TABLE |

- 4 On the computer that will be your Configuration Server and Messaging Server, create an ODBC DSN called CMINVMGR, or name of your choice, pointing to the new INVENTORY database on your Oracle Server. If you do not know how to create an ODBC DSN, contact your Oracle database administrator.



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

Administrator CSDB Editor

The Configuration Server media contains an Administrator installation. Use it to install the CSDB Editor. Refer to the *Administrator Guide* for information on installing and using the CSDB Editor.

Messaging Server

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

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

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

Reporting Server

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

Portal (Optional)

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

3 Installing the Client Automation Agents

At the end of this chapter, you will:

- Understand the system requirements for installing Client Automation agents.
- Know how to customize the installation process.
- Be able to modify the [PROPERTIES] section of the `Install.ini` file in order to customize the behavior of the installation process.
- Be able to modify the [ARGS] section of the `Install.ini` file in order to customize the behavior of the HP Client Automation Application Self-service Manager (Application Self-service Manager).
- Be able to modify the [OBJECTS] section of the `Install.ini` file in order to specify Client Automation objects to be created on the Client Automation agent computer.
- Know how to use the Installation Wizard.
- Know how to remove and repair Client Automation agents using the Installation Wizard and command lines.
- Know how to modify the installation of the Client Automation agents using the Installation Wizard and command lines.
- Know how to use a pre-install script to customize MSI properties that affect installation.
- Know how to use a post-install script to run processes after installing Client Automation agents.

Client Automation Agent Installation

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

- HP Client Automation Application Manager (Application Manager)
- HP Client Automation Application Self-service Manager (Application Self-service Manager)
- HP Client Automation Inventory Manager (Inventory Manager)
- HP Client Automation OS Manager (OS Manager)
- HP Client Automation Patch Manager (Patch Manager)



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

System Requirements

- Microsoft .NET runtime version 1.1 or higher (required for Application Self-service Manager only). The .NET installation program is available in the `\dotnet` folder on the Client Automation agent media. If .NET does not exist on the Client Automation agent computer, the .NET installation program runs automatically. Microsoft .NET requires Microsoft Internet Explorer 5.01 or later.
- TCP/IP connection to a computer running the Configuration Server.
- For Windows 2000, Windows Server 2003, Windows XP, and Windows Vista, you must have administrator rights to the computer to install the Client Automation agents.

Platform Support

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

Client Automation Agent Installation Process

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

- **setup.exe**

Stored in the `\win32` directory on the Client Automation agent media, it accepts any standard Windows Installer command-line parameters and passes them to the Windows Installer service.

You can also create an AIP for network installations.



An Administrative Installation Point is also known as an **Administrative Control Point (ACP)**.

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

```
setup.exe /a TARGETDIR=drive:\targetdirectory /qb
```

The target directory contains `HPCAE-MgmtAppsxx.msi`, the installation folders, `setup.exe`, and any files (such as `Install.ini` or Visual Basic scripts) stored in the same directory as `setup.exe`. Next, copy the `\dotnet` and `\MSI` folders into the target directory.

- **HPCAE-MgmtAppsxx.msi**

This MSI database file is stored in the `\win32` directory on the Client Automation agent media and contains the default configuration information for the installation.

- **Install.ini**

Use `Install.ini` to customize the installation or the Client Automation agent arguments file, or to create or set attributes for Client Automation objects. Settings in `Install.ini` override the defaults stored in `HPCAE-MgmtAppsxx.msi`.

A sample `Install.ini` is available in the `\win32` directory on the Client Automation agent media.

- **args.xml**

The Application Self-service Manager arguments file created from information stored in the [ARGS] section of `Install.ini`. This file, stored in `IDMLIB` on the Client Automation agent computer, controls the

behavior of the Application Self-service Manager. The default directory for IDMLIB is `SystemDrive:\Program Files\Hewlett-Packard\CM\Agent\Lib\`.

- **Pre-install scripts**

(Recommended for experienced users only.)

Use custom Visual Basic scripts to customize MSI properties that affect the installation. For an example of a simple script, see [Using a Pre-Install Script](#) on page 57.

- **Post-install scripts**

(Recommended for experienced users only.)

Use custom Visual Basic, REXX, or Tcl scripts to run processes such as the first Client Automation agent connect. For an example, see [Using a Post-Install Script](#) on page 58.

Preparing Install.ini

Create an installation file, `Install.ini`. Use this file to:

- Customize the installation.
- Customize the Client Automation agents.
- Create or set attributes for Client Automation objects.

Below is a sample `Install.ini`. A description of the three sections (**Properties**, **Args**, and **Objects**) and the attributes in each of these sections follows.



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

The following is a sample `Install.ini`.

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

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

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

;Uncomment and specify the features that you want to install
```

```

;ADDLOCAL=NVDINSTALLRAM,NVDINSTALLRSM,NVDINSTALLRIM,NVDINSTALLRLAE

;Uncomment to show/hide panels in the installation wizard (defaults shown below)
;NVDENABLEUSER=Y
;NVDENABLEIP=Y
;NVDENABLEPROXY=N
;NVDENABLESHORTCUT=Y
;NVDSHORTCUT=Y
;NVDSTARTIMENUICON=Y
;NVDSTARTWMICFGMGR=Y
;NVDDRAMSHORTCUT=N
;NVDDRAMSTARTIMENUSHORTCUT=N
;NVDAMCONNECT="radntfyc localhost radskman cat=prompt,uid=$machine,ulogon=n,hreboot=y,ind=y,
               ask=y,ip=xxx.xxx.xxx.xxx,port=3464,mname=Radia,dname=Software,startdir=SYSTEM,timeout=1
               800,context=m,log=connect_manual.log"

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

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

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

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

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

[Args]
; Tags to be placed into the CM Application Self Service Manager Agent ARGS.XML file
; A value of _NONE_ removes the tag from the file
;
;askconfirm=
;channelname=software
;dataurl=
;enterprisemanagement=
;identification=$USER
;log=connect.log
;logsize=
;logonpanel=
;managerurl=
;providername=Radia

```

```

;redirect=
;resolutionmanager=
;resolutionport=
;sslmanager=
;sslport=
;startdir=
;uioption=

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

```

[PROPERTIES] Section of Install.ini

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



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

Table 3 [PROPERTIES] Section of Install.ini

| Argument | Description |
|-------------|---|
| ADDLOCAL | <p>Specify the features that you want to install on the local hard drive. There is no default for this argument.</p> <p>The features are:</p> <p>NVDINSTALLRAM = Application Manager</p> <p>NVDINSTALLRSM = Application Self-service Manager</p> <p>NVDINSTALLRIM = Inventory Manager</p> <p>NVDINSTALLROM = OS Manager</p> <p>NVDINSTALLPATCH = Patch Manager</p> <p>NVDINSTALLRLAE = Local AIP Extension</p> <p>Note: This must be installed to use the Local AIP Support for the MSI Redirector feature described on page 59.</p> |
| ARPNOREMOVE | <p>Set to 1 to disable the ability to remove the Client Automation agent from the computer using Add/Remove Programs in the Control Panel.</p> <p>For Windows 2000 and XP, the Remove button is disabled.</p> <p>For earlier operating systems, the Client Automation agent will not be listed in Add/Remove Programs in the Control Panel.</p> <p>Note: Setting to 0 will not disable this option due to a Windows Installer issue. If you want to allow your subscribers to remove the Client Automation agent from the computer using Add/Remove Programs, place a semi-colon (;) in front of the ARPNOREMOVE argument in <code>Install.ini</code>.</p> |
| INSTALLDIR | <p>Specify the directory in which to install the Client Automation agent. The default is <code>SystemDrive:\Program Files\Hewlett-Packard\CM\Agent</code>.</p> <p>This value will be overridden if a new directory is specified in the Destination Folder window in the Client Automation Agent Installation Wizard.</p> |

| Argument | Description |
|-----------------------|--|
| NVDENABLEUSER | <p>Indicate whether to show or hide the Set User window in the Installation Wizard.</p> <p>Specify Y (the default) to show the window.</p> <p>Specify N to hide the window.</p> <p>Specify D to show the window, but disable the User Name field. The Create HPCA Application Self-service Manager icon on the desktop check box is still available.</p> <p>Note: If you hide the window, the Create HPCA Application Self-service Manager icon on the desktop check box will no longer be available to your subscribers.</p> |
| NVDENABLEIP | <p>Indicate whether to show or hide the Configuration Server window in the Installation Wizard. The default is Y.</p> |
| NVDENABLE PROXY | <p>Indicate whether to show or hide the Proxy Information window in the Installation Wizard. The default is N.</p> <p>If you want to use a Proxy Server during the Client Automation agent connect, show this window.</p> <p>The information that is specified in the Proxy Information window is stored, by default, in the PROXYINF object in the Client Automation agent computer's IDMLIB directory.</p> |
| NVDENABLE SHORTCUT | <p>Indicate whether to show the Create HPCA Application Self-service Manager icon on the desktop check box in the Set User window. The default is Y.</p> <p>Selecting this check box installs a shortcut on the subscriber's desktop for the Application Self-service Manager.</p> |
| NVDSHORTCUT | <p>Indicate whether to install a desktop shortcut for the Application Self-service Manager on the subscriber's computer. The default is Y.</p> |
| NVDSTART MENUICON | <p>Indicate whether to install an icon in the Start Menu for the Application Self-service Manager on the subscriber's computer. The default is Y.</p> |
| NVDSTARTWMI CFGMGR | <p>Indicates whether to install the shortcuts for WMI. The default is Y.</p> |
| NVDDRAMSHORT CUT | <p>Indicate whether to install a desktop shortcut for the Application Manager on the subscriber's computer. The default is N.</p> |

| Argument | Description |
|-----------------------------|--|
| NVDRAMSTART MENUSHORTCUT | Indicate whether to install an icon in the Start Menu for the Application Manager on the subscriber's computer. The default is N . |
| NVDRAMCONNECT | Specify a command line to run if an Application Manager shortcut is created on the desktop or the Start Menu. There is no default for this argument. |
| NVDMAINTDIR | <p>Specify a directory in which to store the Client Automation agent maintenance files. The default is the <code>MAINT</code> sub-directory of the folder that contains <code>setup.exe</code>.</p> <p>Note: Specify a value only if you want to store maintenance files in a directory other than the default.</p> <p>If files in this directory are more recent than the installation files, they will be copied into the Client Automation agent's <code>IDMSYS</code> directory.</p> |
| NVDLOCAL NOTIFYONLY | If set to Y , the Client Automation agent will allow Notifies from the local host only. The default is N . |
| NVDRADTRAY START | Set to Y to start the System Tray automatically if the Application Manager is selected during the Client Automation agent installation process. The default is N . |
| NVDNOTIFY INTERACT | Set to Y to enable the Notify Daemon to interact with the desktop. The default is N . |
| NVDREDIRECT ORINTERACT | Set to Y to enable the MSI Redirector to interact with the desktop. The default is N . |
| NVDSCHEDULER INTERACT | Set to Y to enable the Scheduler to interact with the desktop. The default is N . |
| NVDPRECAPATH | <p>Specify the fully qualified path and filename of a custom Visual Basic pre-install script. There is no default for this argument.</p> <p>Note: New objects or properties must be defined in <code>Install.ini</code>. You can use a pre-install script to override a value for the object or property, but if you attempt to specify a new object or property in the pre-install script, it will be ignored.</p> <p>For an example of a simple script, see Using a Pre-Install Script on page 57.</p> |

| Argument | Description |
|---------------|---|
| NVDPOSTCAPATH | Specify the fully qualified path and filename of a custom Visual Basic or REXX post-install script. There is no default for this argument. For an example, see Using a Post-Install Script on page 58. |

[ARGS] Section of Install.ini

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

The following is an example of `args.xml`.

```
<?xml version="1.0" ?>
<RADIO_ARGUMENTS>
<ARGUMENTS><CHANNELNAME>software</CHANNELNAME>
<IDENTIFICATION>jsmith</IDENTIFICATION>
<PROVIDERNAME>radio</PROVIDERNAME>
<RESOLUTIONMANAGER>10.10.10.1</RESOLUTIONMANAGER>
<LOG>connect.log</LOG>
<RESOLUTIONPORT>3464</RESOLUTIONPORT>
<ROOT_CATALOG_NAME>All Software</ROOT_CATALOG_NAME>
</ARGUMENTS>
</RADIO_ARGUMENTS>
```



The XML tags (arguments) that are described in this section are not case-sensitive when you type them in `Install.ini`. However, they will be automatically converted to uppercase in `args.xml`.

If you are using the Application Manager, any of the parameters in the [ARGS] section can be added to the RADSKMAN command line.

Table 4 [ARGS] section of Install.ini

| Argument | Mandatory or Optional | Description |
|----------------------|-----------------------|--|
| askconfirm | Optional | Controls the display of a confirmation message to your subscribers. For example, some instances in which a confirmation message might display are: A reboot is required. There is insufficient disk space during deployment. A data download is interrupted. The default is Y . |
| channelname | Mandatory | The CSDB Domain from which applications are retrieved. The default is SOFTWARE . |
| default_catalog | Optional | Set the default catalog that is selected when the Application Self-service Manager starts. There is no default for this argument. |
| default_catalog_only | Optional | Set to Y to make only the Default_catalog available when the Application Self-service Manager starts. The default is N . |
| identification | Optional | Identifies the Client Automation agent to the Configuration Server by defining the value for the ZUSERID variable in the ZMASTER object. The default is \$USER . This value will be overridden if a different User Name is specified in the Set User window in the Agent Installation Wizard. If you do not want this value to be modified, set NVDENABLEUSER=N in the [PROPERTIES] section of <code>Install.ini</code> . \$MACHINE : The Client Automation user ID is the name of the subscriber's computer. \$USER : The Client Automation user ID is the logon ID for the subscriber currently logged on. CUSTOM : literal custom specification. |
| log | Optional | Specifies the name of the log stored in <code>IDMLOG</code> . <code>IDMLOG</code> is specified in <code>NVD.INI</code> . The default is <code>Connect.log</code> . The default location of <code>NVD.INI</code> is |

| Argument | Mandatory or Optional | Description |
|-------------------|-----------------------|---|
| | | <i>SystemDrive:\Program Files\Hewlett-Packard\CM\Agent\LIB.</i> |
| logsize | Optional | Specifies (in bytes) the size of the log file. The default is 1000000 . When the logsize is reached, a backup file (.bak) is created. By default, this file is <code>connect.bak</code> . If a backup file already exists, it will be overwritten. |
| logonpanel | Optional | Controls the display of the logon panel. The default is Y . |
| managerurl | Optional | Specifies the address (in the form <i>http://hostname:port/nvdurl</i>) of the Configuration Server to be used for HTTP object transfer. There is no default for this argument. |
| providername | Mandatory | The name of the Configuration Server, as was set during its installation. The default is radia . This is used to name the folder below the STARTDIR on the Client Automation agent computer. See startdir on page 43 for more information. |
| redirect | Optional | <i>Used for the Application Self-service Manager only.</i> Specifies an alternate start-up file (<i>filename.xml</i>) that can be accessed via a network path or URL. There is no default for this argument. If the redirect tag is set in <code>args.xml</code> , the Application Self-service Manager uses the properties that are specified in the alternate file. |
| resolutionmanager | Mandatory | The IP address of the Configuration Server. The Configuration Server name can also be used. There is no default for this argument. This value will be overridden if a different IP address is specified in the Configuration Server window in the Agent Installation Wizard. To prevent this value being modified, set NVDENABLEIP=N in the [PROPERTIES] section of <code>Install.ini</code> . |

| Argument | Mandatory or Optional | Description |
|-------------------|-----------------------|--|
| resolutionport | Mandatory | <p>The port for the Configuration Server. There is no default for this argument.</p> <p>This value will be overridden if a different port is specified in the Configuration Server window in the Agent Installation Wizard.</p> <p>To prevent this value being modified, set NVDENABLEIP=N in the [PROPERTIES] section of <code>Install.ini</code>.</p> |
| root_catalog_name | Mandatory | <p>Use this to customize the name of the root catalog display name. There default is All Software.</p> |
| sslmanager | Optional | <p>The address of the Configuration Server that is to be used for SSL communications. There is no default for this argument.</p> <p>If you want self-maintenance to use SSL communications, append : :SM to the end of the specified IP address or host name, as in sslmanager=hostname: :SM.</p> <p>Warning: Use the : :SM switch with the following caveat in mind: the file (<code>cacert.pem</code>) that contains the CA root certificates cannot be maintained. If the corresponding CA root certificate for the certificate in use by the Configuration Server should ever become expired, revoked, or corrupt, it will result in disabling SSL communications to the Configuration Server.</p> |
| sslport | Optional | <p>The TCP/IP port (usually 443) on which the SSL Manager task is listening. There is no default for this argument. The sslport specification takes the form <code>sslport=port</code>.</p> |
| startdir | Optional | <p>The starting IDMLIB directory (by default, <code>SystemDrive:\Program Files\Hewlett-Packard\CM\Agent\Lib\</code>). The default is \$USER.</p> <p>\$MACHINE: The current computer name.</p> <p>\$USER: The user ID of the subscriber that is currently logged on.</p> <p>CUSTOM: literal custom specification. Type</p> |

| Argument | Mandatory or Optional | Description |
|----------|-----------------------|---|
| | | startdir= <i>foldername</i> . If the folder name contains embedded spaces, enclose the entire name in quotation marks (“ ”). Note: Do this to distribute a set of applications that will be shared by all subscribers of a multi-user computer. |
| uioption | Optional | Controls the display of the status window. The default is N . |

[OBJECTS] Section of Install.ini

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

Table 5 [OBJECTS] section of Install.ini

| Argument | Description |
|------------------|---|
| ZMASTER_ZDSTOCK | The port setting for the Configuration Server. The default is 3464 . |
| ZMASTER_ZIPADDR | The IP address for the Configuration Server. There is no default for this argument. |
| ZMASTER_ZNTFPORT | The port on which the Client Automation agent’s Notify daemon is “listening.” The default is 3465 . |
| ZMASTER_ZNTFYSEC | This attribute allows a Notify operation to execute programs from the <code>IDMSYS</code> directory only. This is used for security during Notify operations. The default is Y . |
| ZMASTER_ZTIMEO | The duration (in seconds) that the Client Automation agent will wait for a response from the Configuration Server before it times out. The default is 240 . Valid values are numerals from 0 to 3200. |

| Argument | Description |
|-------------------|--|
| ZMASTER_ZTRACE | Indicates whether communications buffer information will be included in the log; also generates unique logs for create methods. The default is N . Y enables Communication and Method Tracing. S enables Communication summary information; Method Tracing is not enabled. N disables Communication Tracing and Method Tracing. |
| ZMASTER_ZTRACEL | The level of tracing that is generated in the Client Automation agent log files. The default is 040 . Valid values are 0 to 999, where 0 = minimal tracing, 40 is acceptable for most activity, and 999 = maximum tracing. |
| ZMASTER_ZUSERID | The subscriber's user ID. The default is the name of the user who is logged on to the computer. |
| ZMASTER_ZVRFYUID | Specify Y to verify the user ID that was sent by the Notify command. This verification uses the ZUSERID field from the Client Automation agent's ZMASTER object. The default is N . |
| PROXYINF_USEPROXY | Indicates whether a proxy server is used when connecting to the Configuration Server. The default is N . |
| PROXYINF_DISCOVER | <i>For use with Microsoft Internet Explorer.</i> Set to the proxy address and port that your web browser is using. The default is N . |
| PROXYINF_PROXADDR | The IP address and port number of your proxy server. The default is xxx.xxx.xxx.xxx:1080 . |
| RADSETUP_COP | Set to Y to enable Client Operations Profiles. The default is N . |

Installing the Client Automation Agents

The Client Automation agent installation can be initiated by one of the following methods.

- a *command line*

See [Installing the Client Automation Agent from a Command Line](#) starting on page 46.

- a *logon script*

See [Initiating the Client Automation Agent Installation from a Logon Script](#) starting on page 49.

- the *Portal*

This is recommended for mass rollouts in Windows environments. Refer to the *Portal Guide* for more information.

After initiating the installation, the Client Automation agent installation program runs.

This section describes some of the ways that you can initiate the Client Automation agent installation, and then describes the standard Agent Installation Wizard.

Installing the Client Automation Agent from a Command Line

Before performing an installation from a command line, determine:

- Which Client Automation agents to install (See [Referencing the Client Automation Agents](#), below), and
- How the Client Automation agent installation program will be made available to your subscribers. This can be done via: a web page, an FTP site, a mapped drive, a CD-ROM, and e-mail,

Then pass the necessary arguments on a command line. The arguments are detailed in the section, [Specifying the Client Automation Agent Features to Install](#), starting on page 47, as well as in [Table 6](#) on page 47 and [Table 7](#) on page 47.

Referencing the Client Automation Agents

When specifying the arguments on the command line, reference the Client Automation agents as listed below.

- NVDINSTALLRAM to install the Application Manager.
- NVDINSTALLRSM to install the Application Self-service Manager.
- NVDINSTALLRIM to install the Inventory Manager.
- NVDINSTALLROM to install the OS Manager.
- NVDINSTALLPATCH to install the Patch Manager.

Specifying the Client Automation Agent Features to Install

To specify the features that you want to install, use the appropriate feature state argument, as described in the [Table 6](#) below.

Table 6 Client Automation agent feature state arguments

| Specify: | Action |
|----------|--|
| ADDLOCAL | Type a comma-delimited list of features that you want set to “Will be installed on local hard drive.” |
| REMOVE | <p>Type a comma-delimited list of features that you want set to “Entire feature will be unavailable.”</p> <p>This removes the features only, not the product. Therefore, if you use the REMOVE property and type each of the feature names, the core product will still be stored on the computer.</p> <p>To remove the product, type REMOVE=ALL.</p> |

Additional Command Line Arguments

Additional arguments that you can pass to the installation program on the command line are described [Table 7](#) below.

Table 7 Command line arguments

| Sample | Action |
|----------------------------|--|
| /qn | <p>Performs a silent installation.</p> <p>Note: A silent installation is one that takes place without a user interface. This might also be referred to as a “quiet installation,” or an “unattended installation.”</p> |
| /qb | Displays the progress bar only during the installation. |
| /L*v drive:\install.log | <p>Creates a detailed Windows Installer log.</p> <p>Note: Using this option could impact the performance of the installation.</p> |

| Sample | Action |
|---|--|
| <code>/a TARGETDIR=drive: \targetdirectory</code> | <p>Creates a Windows Installer AIP in the specified target directory.</p> <p>Note: A Windows Installer AIP is also known as an ACP.</p> <p>The target directory contains <code>RADIA.MSI</code>, the installation folders, <code>setup.exe</code>, and any files (such as <code>Install.ini</code> and Visual Basic scripts) that are stored in the same directory as <code>setup.exe</code>.</p> <p>After you have created the AIP, you can run <code>setup.exe</code> and pass the command-line parameters. This starts the Windows Installer and passes the specified parameters to it.</p> |
| <code>NVDINIFILE=path \INIfilename</code> | <p>To rename the installation INI file, pass this parameter to the command line. Be sure to include the fully qualified path.</p> <p>By default, the installation program refers to <code>Install.ini</code>, which is located in the current directory.</p> |
| <code>INSTALLDIR=</code> | Specify the installation directory. Use quotation marks if the path contains spaces. |

If you initiate a Client Automation agent installation with a command line that does not contain the silent installation argument (`/qn`), the Client Automation Agent installation program will open. See [Using the Agent Installation Wizard](#) on page 50 for more information.

Examples

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

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

The following is an example of a command line that will install the Application Manager and the Application Self-service Manager.

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

The arguments in this command line, and others, are described in [Specifying the Client Automation Agent Features to Install](#) on page 47 and [Table 7](#) on page 47.

Initiating the Client Automation Agent Installation from a Logon Script

You can use a logon script on a Windows 2000, Windows 2003 Server, Windows XP Professional, or Netware agent to automate the Client Automation agent installation.



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

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

Sample Logon Script

```
:begin
@echo off

if exist C:\progra~1\Hewlett-Packard\CM\Agent\LIB\zmaster.edm
goto skipinst

    start setup.exe /qn

:skipinst

if exist C:\progra~1\Hewlett-Packard\CM\Agent\lib\zmaster.edm
goto skipinst
```

To determine if the Client Automation agents already exist, the script checks for the ZMASTER object (ZMASTER.EDM) in its default location on the computer. If ZMASTER:

- Exists, the script skips the installation.
- Does not exist, the Client Automation agent installation program launches.



The ZMASTER object begins the resolution process and is the first object to be exchanged during the Client Automation agent “connect.”

In the sample logon script, the command, **start setup.exe /qn**, instructs the program to perform a silent installation of the Client Automation agents.



Modify this script to reflect your organization's needs.

If the command line does not contain the silent installation arguments, the graphical Client Automation Agent installation program opens. See [Using the Agent Installation Wizard](#) below for more information.

Using the Agent Installation Wizard

If you start a Client Automation agent installation without the arguments for a silent installation, the Agent Installation Wizard opens. The following steps describe the standard installation procedure. These steps can vary based on `Install.ini` or any arguments passed when running the installation.

To install Client Automation agents using the Installation Wizard

- 1 From the folder containing the Client Automation agent installation files, run `setup.exe`.



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

The Agent Installation Wizard opens.


- 2 Click **Next**. The License Agreement window opens.
- 3 After reading and accepting the license agreement, click **Next**.

The Destination Folder window opens.

The default location for the Client Automation agents is
`SystemDrive:\Program Files\Hewlett-Packard\CM\Agent`.

If you want to select a different destination for the Client Automation agent, click **Browse** and navigate to the appropriate destination folder. This overrides the value set for `INSTALLDIR` in `Install.ini`.

- 4 Click **OK** to continue.
- 5 Click **Next**. The Set User window opens.
- 6 In the User Name text box, type the name of the subscriber for whom you are installing the Client Automation agents. This overrides the value set for `IDENTIFICATION` in `Install.ini`.
- 7 Select the **Create HPCA Application Self-service Manager icon on the desktop** check box if necessary.

- 8 Click **Next**. The Configuration Server window opens.
- 9 In the IP Address text box, type the IP address for the Configuration Server. This overrides the value set for RESOLUTIONMANAGER in `Install.ini`.
- 10 In the Port text box, type the port number. This overrides the value set for RESOLUTIONPORT in `Install.ini`.
- 11 Click **Next**. The Select Features window opens.
- 12 Click  to select the features that you want to install.

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




Install only the Client Automation agents for which you are licensed.

- 13 From the shortcut menu, select an installation option. These options are described in [Table 8](#) below.

Table 8 Feature Settings for Client Automation Agents

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



If you want to set the same options for all of the features, you can click  **HPCA Agent** and select the appropriate option to apply the setting to all features.

Click **Disk Cost** to see an overview of the disk space needed for the installation.

14 Click **Next**.

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

15 Click **OK**.

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

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

17 Click **Next**.



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

The Ready to Install the Application window opens.

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

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

When the installation is done, successful installation window opens.

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

Removing the Client Automation Agents

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

Using the Installation Wizard to Remove Client Automation Agents

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



To remove features of the Client Automation agent, use the Modify option on the Application Maintenance window. This is discussed in [Modifying the Client Automation Agent Installation](#) on page 55.

To remove Client Automation agents using the Installation Wizard

- 1 From the folder containing the Client Automation agent installation files, double-click **setup.exe**. The Application Maintenance window opens.
- 2 Select the **Remove** option.
- 3 Click **Next**. The Client Automation Agent Uninstall window opens.
- 4 Click **Remove**. The files for all Client Automation agents are removed from the computer.

The Client Automation agent has been successfully uninstalled window opens.

- 5 Click **Finish**.

Using a Command Line to Remove Client Automation Agents

This section describes how to use a command line to remove Client Automation agents.

To remove Client Automation agents using a command line

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

SETUP.EXE REMOVE=ALL

See [Installing the Client Automation Agent from a Command Line](#) on page 46 for additional arguments.

or

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

Reference the features for the Client Automation agent installation as follows:

- Application Manager NVDINSTALLRAM
- Application Self-service Manager NVDINSTALLRSM

- Inventory Manager NVDINSTALLRIM
- OS Manager NVDINSTALLROM
- Patch Manager NVDINSTALLPATCH

Example

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

```
SETUP.EXE REMOVE=NVDINSTALLRSM,NVDINSTALLRAM /qn
```



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

Repairing the Client Automation Agents

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

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

Using the Installation Wizard to Repair Client Automation Agents

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

To repair Client Automation agents using the Installation Wizard

- 1 From the folder containing the Client Automation agent installation files, double-click **setup.exe**.
The Application Maintenance window opens.
- 2 Select the **Repair** option.
- 3 Click **Next**. The Ready to Repair the Application window opens.

- 4 Click **Next**. When the repair is done, the Client Automation agent has been successfully installed window opens.
- 5 Click **Finish**.

Using a Command Line to Repair Client Automation Agents

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

To repair Client Automation agents using a command line

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

```
msiexec /f HPCAE-MgmtAppsxx.msi
```



In the above command line, the `xx` is a placeholder for the version of the Management Applications software release; be sure to replace this with the appropriate version number.

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

Modifying the Client Automation Agent Installation

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

Using the Installation Wizard to Modify the Client Automation Agent Installation

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

To modify the installation of Client Automation agents using the Installation Wizard

- 1 From the folder containing the Client Automation agent installation files, double-click **setup.exe**. The Application Maintenance window opens.
- 2 Select the **Modify** option.
- 3 Click **Next**. The Select Features window opens. See [Using the Agent Installation Wizard](#) on page 50 for information about how to use this window.
- 4 Click **Next**. The Ready to Modify the Application window opens.
- 5 Click **Next**. The Client Automation agent has been successfully installed window opens.
- 6 Click **Finish** to close the installation program.

Using a Command Line to Modify the Client Automation Agent Installation

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

To modify the installation of Client Automation agents using a command line

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

SETUP.EXE FeatureStateArgument=feature1 , feature2

Table 9 Client Automation agent feature state arguments

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

Reference the features for the Client Automation agent installation as follows:

- Application Manager NVDINSTALLRAM
- Application Self-service Manager NVDINSTALLRSM
- Inventory Manager NVDINSTALLRIM
- OS Manager NVDINSTALLROM
- Patch Manager NVDINSTALLPATCH

Example

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

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

See [Installing the Client Automation Agent from a Command Line](#) on page 46 for additional arguments.

Using a Pre-Install Script

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



Be sure to use the NVDPRECAPATH argument to specify the fully qualified path and file name of a custom Visual Basic pre-install script in `Install.ini` or on the command line. See the description of NVDPRECAPATH in [Table 3](#) on page 37.

Here is a sample pre-install script:

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

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

```
Option Explicit
```

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

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

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

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



New objects or properties must be defined in `Install.ini`.

You can use a pre-install script to override a value for the object or property, but if you attempt to specify a new object or property in the pre-install script, it will be ignored.

To override property settings or attributes for objects



Be sure to type the name of the property or the object and its attribute such as `NVDOBJZMASTER_ZDSTOCK` in uppercase letters.

- Use the prefix `NVDARG` to override property settings.

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

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

- Use the prefix `NVDOBJ` to override object attributes.

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

```
Session.Property("NVDOBJZMASTER_ZDSTOCK")="3462"
```

Using a Post-Install Script

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



Be sure to use the NVDPOSTCAPATH argument to specify the fully qualified path and filename of the custom Visual Basic or REXX post-install script in `Install.ini` or on the command line. See the description of NVDPOSTCAPATH in [Table 3](#) on page 37. For example, if you want to run a script called `redstart.rex`, uncomment and set `NVDPOSTCAPATH=C:\Progra~1\Hewlett-Packard\CM\Agent\radstart.rex`

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

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

```

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

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

```

Local AIP Support for the MSI Redirector



Windows 2000, Service Pack 4 Note

In order for the Local AIP driver to work, the Microsoft Windows 2000 Rollup 1 cumulative patch must be installed.

The MSI Redirector is a specialized, local-host HTTP server that accepts and satisfies file requests that are made during an MSI installation. On receiving the HTTP request from MSI, the Redirector retrieves the file from its local cache (if it exists); if the file is not in its local cache, the Redirector requests the file from an upstream Configuration Server or Proxy Server. This process requires that the MSI installation supports HTTP, although some vendors (including Microsoft) have removed HTTP support from their product installations. Without HTTP support, the MSI installation will not be able to directly request the files from the MSI Redirector.

In order to continue to use the MSI Redirector, a level of redirection now exists at the local file-system level. When using this method, MSI is told that the AIP is local, and requests the files directly from the file system. This request is captured and forwarded to the MSI Redirector, which satisfies the request in its usual way. The file is then placed in the defined local AIP where MSI can process it. The local AIP is temporary; it is removed after the installation is completed.

To enable using the Local Application Installation Point

- 1 Use the CSDB Editor to navigate to the MSI Resources (MSI) Class in the SOFTWARE Domain. Each MSI application will have an MSI instance and an IDX instance.
- 2 Right-click the MSI instance and select **Edit Instance**.
- 3 Set **MSIDRIVR (Use Local AIP [Y/N])** to **Y**.



If MSIDRIVR is not in your database, create it in the MSI Resources (MSI) Class as a 1-byte variable with a description of **Use Local AIP [Y/N]**.

HP recommends backing up your database before making changes to a Class template.

For information on editing Class templates, refer to the *Administrator Guide*.

- 4 Click **OK**.
- 5 Click **Yes** to confirm the changes.
- 6 Configure the SETTINGS.LOCALAIP variable in COPs to control the destination of the local AIP folder on the Client Automation agent desktop. For example, **C:\localaip**.



HP recommends keeping your LOCALAIP as short as possible to accommodate AIPs that have deep directory structures.

Internet Proxies

Internet proxies are put in place by companies for a variety of reasons. HP Client Automation can detect when an internet proxy is being used. It stores the proxy's address in `PROXYINF.EDM`, which is in the Client Automation

agent computer's `IDMLIB` directory, thereby allowing the Client Automation agent authority to pass through the proxy.

You must enable the Client Automation agent to discover and use internet proxies by setting

USEPROXY=Y and **DISCOVER=Y**

in the Client Automation agent `PROXYINF.EDM` object.

The next time that the Client Automation agent connects to the Configuration Server it will use the internet proxy that is specified.

4 The AUDIT Domain

At the end of this chapter, you will:

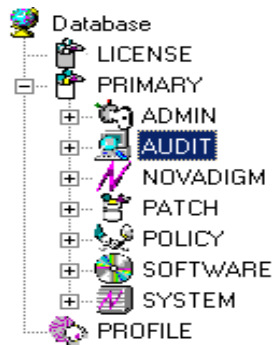
- Understand the AUDIT Domain in the HP Client Automation Configuration Server Database (CSDB).
- Understand the HP Client Automation Inventory Manager (Inventory Manager) database.

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

The AUDIT Domain in the CSDB

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

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



AUDIT Domain Defined

The AUDIT Domain is structured very much like the SOFTWARE Domain. The next figure shows its tree structure in the CSDB Editor.

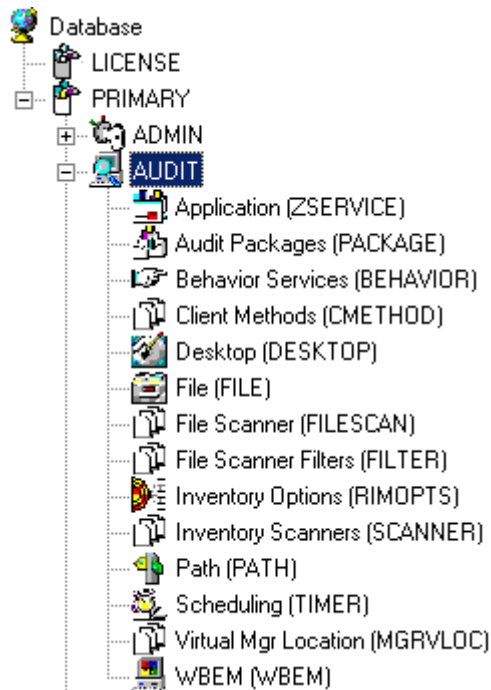


Table 10 below describes the classes in the AUDIT Domain.

Table 10 AUDIT Domain

| Class | Description |
|---------------------------------|--|
| Application (ZSERVICE) | These are sample services distributed with the Inventory Manager. The AUDIT.ZSERVICE instance is connected to a policy instance. A policy instance can be an instance of the Users, Departments, or Workgroups class. It can also be a customer-defined class within the POLICY Domain. Each of the sample ZSERVICE classes is connected to the PACKAGE instances. |
| Audit Packages (PACKAGE) | Defines what information to collect, and then what actions to take. These packages would contain various audit components. A good example is an audit of running services on a desktop. The AUDIT.ZSERVICE instance must contain a connection to an AUDIT.PACKAGE instance. |

| Class | Description |
|--------------------------------------|--|
| Behavior Services (BEHAVIOR) | Defines instances that enable the execution of auditing on the agent. Normally, there is no need to add or modify instances in this class. |
| Client Methods (CMETHOD) | This class is used to configure method points for Tel inventory scans. The base instance of the SCANNER Class is connected to the CMETHOD.INV_FULL instance. This instance can be used for all inventory scans defined in the SCANNER Class. |
| Desktop (DESKTOP) | This class is reserved for future use. |
| File (FILE) | Defines file scans, such as auditing system DLLs. |
| File Scanner (FILESCAN) | For UNIX devices only, persistent component class used to configure an inventory scan. Adding File Scanner components to an audit package creates instances of the FILESCAN Class. |
| File Scanner Filters (FILTER) | For UNIX devices only, persistent component class used to configure an inventory scan. Adding FILE Scanner Filters components to an audit package creates instances of the FILTER Class. |
| Inventory Options (RIMOPTS) | Contains the attributes that offer options that control an inventory management task. For additional information, refer to the RIMOPTS Class on page 67. |
| Inventory Scanners (SCANNER) | This persistent component class is used to configure an inventory scan. Create instances of the SCANNER Class by adding Inventory Scanners components to an audit package. |
| Path (PATH) | This class stores the drive and directory required to install a resource. Packages can be relocated by updating instances of this class. |
| Registry (REGISTRY) | This class uses WMI to obtain a Registry scan of a Windows machine. Create instances of the REGISTRY Class to run scans of the Windows Registry and obtain a Registry Scan report. |

| Class | Description |
|---------------------------------------|--|
| Scheduling (TIMER) | This class contains the instances that enable the CM administrator to set a timer on end users' computers. One or multiple auditing services can be processed whenever the timer expires. |
| Virtual Mgr Location (MGRVLOC) | This class is used to specify the initial path for files being transferred to the Configuration Server during a FILE audit. |
| WBEM (WBEM) | This class contains instances that define Inventory Manager scans of WMI classes. These can include any class in the WMI database such as Win32_Services. This example would provide information on Windows NT or Windows 2000 services. |

RIMOPTS Class

The RIMOPTS Class is also known as the Inventory Options Class. This class contains the attributes that control an inventory management task. [Table 11](#) below describes these attributes.

Table 11 RIMOPTS Class

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

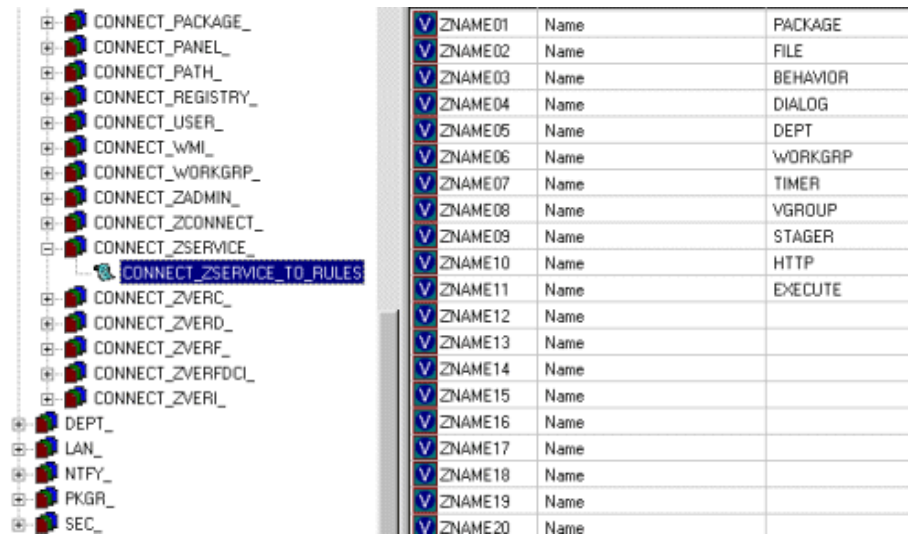
| Attribute | Usage |
|-----------------|--|
| RUNEXEC | <p>This string indicates what actions the Inventory Manager will take upon connection:</p> <ul style="list-style-type: none"> • Select I to invoke collection of information when the service is installed • Select U to invoke collection of information when the service is updated. • Select V to invoke collection of information when the service is verified. <p>The default settings are I and U.</p> |
| ZSVCTYPE | Contains code that is used internally by the Inventory Manager agent. In all cases, this value should remain I . |
| NAME | Contains the friendly name of the instance. It is the name displayed for the instance in the tree view of the CSDB Editor. |

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

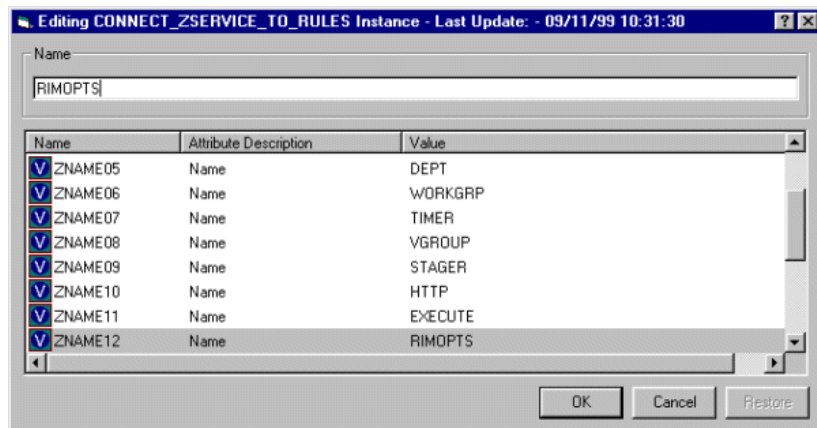
Prior to beginning any tasks using the Inventory Manager, you must enable the drag-and-drop feature for the newly created RIMOPTS Class instances. For additional information about editing instances, refer to the *Administrator Guide*.

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

- 1 Open the Admin CSDB Editor and go to **PRIMARY** → **ADMIN** → **Name Lists (8) (ZLIST)** → **CONNECT_** → **CONNECT_ZSERVICE_**
- 2 Double-click on **CONNECT_ZSERVICE_TO_RULES**.



- 3 The Editing Instance dialog box opens.



- 4 Set the value of the **ZNAME n** attribute to **RIMOPTS**.

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

REGISTRY Class

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

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

- **RPTCLASS** – The Report Class Name in RIM.
- **FORMAT** – The Output format- requires REGISTRY (do not change).
- **DEPTH** – Defines the levels below the current subkey to scan.

Table 12 below summarizes the attributes and values for the Registry Class instances. Attributes in bold are new to this class (not in the WBEM class).

Table 12 Registry Class Instance Attributes

| Attribute | Description | Default Value | Valid Values |
|-----------|--|--------------------------------------|--|
| ACTION | Report Flags (I, N, C, D, S, D, C) | YYYYXXN | Y, X, or N for each flag. |
| NAMESPACE | Name Space | root\default | root\default – Do not change. |
| CLASS | WBEM Class | StdRegProv | StdRegProv – Do not change. |
| RPTCLASS | Report Class Name | Registry | A valid table name. If blank “StdRegProv” will be used. |
| PROPERTY | Registry hive. | HKEY_LOCAL_MACHINE | Any Windows registry hive: HKEY_CLASSES_ROOT HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_CURRENT_CONFIG HKEY_DYN_DATA |
| CNDITION | Registry subkey | SOFTWARE\Microsoft\Internet Explorer | Any Windows registry subkey. |
| FORMAT | Output format. | REGISTRY | REGISTRY – Do not change. |

| Attribute | Description | Default Value | Valid Values |
|-----------|--|---------------|---|
| DEPTH | Starting at the registry subkey named in the CNDITION attribute, depth specifies the number of descendent key levels to include in the scan. | 0 | 0, -1, or <i>n</i> <ul style="list-style-type: none"> Set to 0 to only scan current subkey. Set to -1 to scan all subkey levels. Set from 1 through <i>n</i> to scan the current subkey and the specified number of subkey levels deep. |
| OUTPUT | Output Object Name | WBEMAUDT | WBEMAUDT |
| TYPE | Scan Type (WBEM) | WBEM | WBEM – Do not change. |
| NAME | Friendly Name | Default | Friendly name for this instance displayed in the Admin CSDB Editor. |

Implementing Registry Scans

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

- 1 Create an AUDIT.PACKAGE instance for the registry scan.
- 2 Right-click on the newly created AUDIT.PACKAGE instance and select **Add Component** from the shortcut-menu.
- 3 Use the Add Component dialog to both create and edit a new AUDIT.REGISTRY instance in a few steps:
 - a Use the Available Components drop-down list to select **Registry**.
 - b In the New Component Name text box type an instance name for the new registry scan.
 - c Click **Add + Edit**.

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

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











































































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

- 4 Connect the audit package to an audit service.
- 5 Entitle the audit service for the registry scan to the appropriate machines or users.
- 6 The registry scan service is deployed during the first connection to an entitled agent. Upon the next connection, the registry scan inventory is collected and passed to the Messaging Server, which posts it to the ODBC database for inventory.
- 7 View the Registry Scan report from the Reporting Server.

Inventory Database Tables

The inventory reporting database includes the tables shown in [Figure 3](#) on page 73, among others.

Figure 3 Standard Inventory Database – Tables.

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

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

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

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

5 Software and Hardware Auditing

At the end of this chapter, you will:

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

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

Auditing Types

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

The Inventory Manager allows for three types of audits:

- File auditing
- WBEM auditing
- Hardware auditing

File Auditing

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

See [What Happened during the Scan](#) on page 101 for additional information on the `RIMFSCAN` and the `RIMDIFF` methods.

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

Table 13 `AUDIT.FILE` Class Instances

| Attribute | Description and Examples |
|-----------|---|
| SCANFOR | Indicate a fully qualified path and file name to search for. Wildcards are permitted. <i>Drive:\WinNt**.dll</i> |
| ACTION | The <code>RIMDIFF</code> method performs actions on the files discovered on the user's computer during the agent connect. |

| Attribute | Description and Examples |
|-----------|--|
| | <ul style="list-style-type: none"> • Y configures RIMDIFF to perform the action. • N configures RIMDIFF to not perform the action. <p>The first four flags determine <i>when</i> to report that the files were found:</p> <p>Report on: Initial, New, Changed, Deleted</p> <ul style="list-style-type: none"> • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan. • Deleted means that the file was found during the previous scan. The file is not present for the current scan. <p>The last three flags control the <i>actions to take</i> on the files detected during the current scan.</p> <p>Action to take on discovery: Send, Delete, Custom</p> <ul style="list-style-type: none"> • Send means to send the files to the Configuration Server and store them in the location indicated by the ZRSCVLOC attribute (see ZRSCVLOC in this table). • Delete means to delete the files from the user's computer. • Custom means to execute the method indicated in the CUSTOM attribute. <p>YYYYNYN – Report whenever encountered and delete the files.</p> <p>NNYYNNN – Report when changed or deleted and take no action.</p> <p>NYYYNYN – Report when the files are new or changed. Then send and delete the files.</p> |
| OUTPUT | Output object name. |

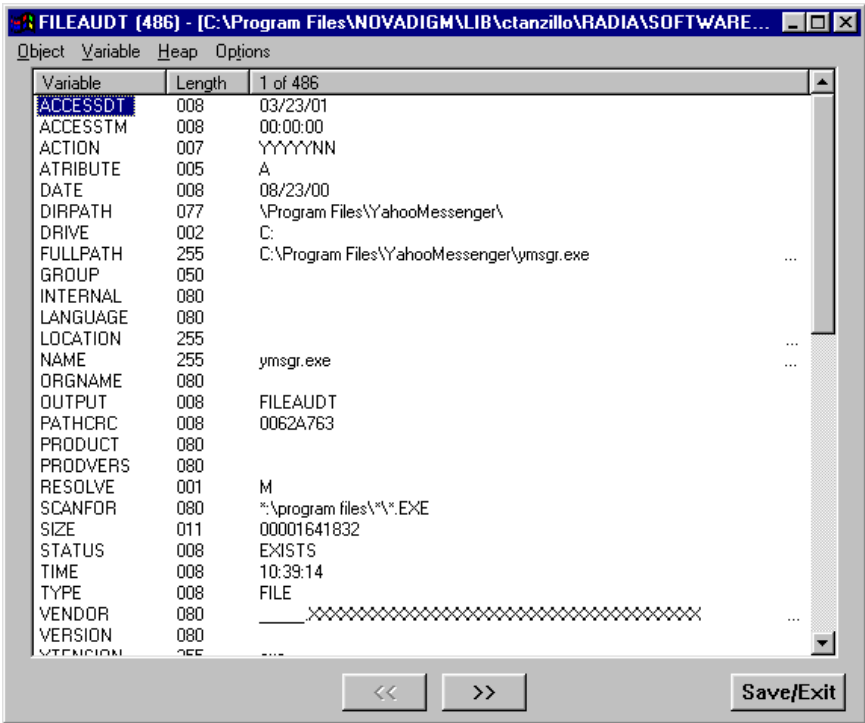
| Attribute | Description and Examples |
|-----------|---|
| TYPE | Scan different file locations. Available scans are Behavior Services, Desktop, File, Path, Registry, and WBEM. File. |
| GROUP | Optional way to identify a set of scan results. This maybe useful for querying and reporting on the audited files from the database where audit results can be stored. Games, MPEGs. |
| ZVERINFO | <p>Collect extended information.</p> <ul style="list-style-type: none"> • Set the value to 1 to collect additional information for a file. • Set the value to 0 to not collect additional information. <p>In order for this data to be collected, the associated attribute must exist in the AUDIT.FILE class template.</p> <p>You can limit the scan to only those files that have some particular values in their extended information. You do so by supplying a value (either 1 or 0) for any of the associated attributes in an AUDIT.FILE instance. This causes the scan to be filtered. Only those files whose extended information data element contains the value you specify in its associated attribute will be scanned.</p> <p>Extended file information consists of one ore more of the following data elements. The associated attribute name for the data element is in parentheses:</p> <ul style="list-style-type: none"> • (VENDOR) – The seller of the file/product • (PRODUCT) – The name of the item for which the file is a part. • (PRODVERS) – The version of the product which the file is a part. • (ORGNAME) – The name of the organization. • (INTERNAL) – The internal data element encoded in the file. • (VERSION) – The version of the file. • (LANGUAGE) – The language of the file. |

| Attribute | Description and Examples |
|-----------|---|
| ZRSCSTYP | Server file type. This can be either Binary or Text. The administrator does not set this. |
| ZRSCMFIL | Manager directory location. |
| ZRSCVLOC | The location on the 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. <i>SystemDrive:\Data\&(ZOBJPID)\&(name)</i> |
| ZRSCMMEM | PDS member name. This field is optional. |
| PRODUCT | The product name. See ZVERINFO on page 78 for more detail. |
| PRODVERS | The product version. See ZVERINFO on page 78 for more detail. |
| ORGNAME | The organization name. See ZVERINFO on page 78 for more detail. |
| INTERNAL | The internal data element encoded in the file. See ZVERINFO on page 78 for more detail. |
| VERSION | The version of the file. See ZVERINFO on page 78 for more detail. |
| LANGUAGE | The language of the file. See ZVERINFO on page 78 for more detail. |
| VENDOR | The product vendor. See ZVERINFO on page 78 for more detail. |
| ZRSCCRC | Resource CRC. |

| Attribute | Description and Examples |
|-----------|--|
| ZCRCINFO | <p>Collect file CRC. [Y/N] Default is N.</p> <ul style="list-style-type: none"> Set the value to Y to collect CRC information for a file. Set the value to N to not collect CRC information. If blank, defaults to N. <p>Caution: Collecting file CRC information can dramatically extend the time it takes to collect information on the target machine.</p> |
| ZMD5INFO | <p>Collect file MD5 information. [Y/N] MD5 information is a 32-character value that can be used to uniquely identify a file based on its content. Default is N.</p> <ul style="list-style-type: none"> Set the value to Y to collect MD5 information for a file. Set the value to N to not collect MD5 information. If blank, defaults to N. <p>Caution: Collecting MD5 information can dramatically extend the time it takes to collect information on the target machine.</p> |
| ZRSCOBJN | Persistent object name. |
| ZRSCPADM | Administrator ID. |
| ZRSCSRC | Resource Source, i.e., Publisher. |
| ZINIT | Not applicable at this time. |
| NAME | Not applicable at this time. |
| LOCATION | Not applicable at this time. |

Use the Agent Explorer to view the FILEAUDT object results as shown in [Figure 4](#) on page 81.

Figure 4 FILEAUDT Object



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

Table 14 FILEAUDT Object

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

| Attribute | Description |
|-----------|---|
| DATE | The date of the most recent modification to this file. |
| DIRPATH | The directory path of the file. |
| DRIVE | The system drive location of the file. |
| FULLPATH | Fully qualified path and file name of the file. |
| PATHCRC | A unique number that indicates the CRC path used for differencing. |
| RESOLVE | The value of M indicates that the Configuration Server resolves each heap of the FILEAUDT object individually. This value cannot be modified. |
| SIZE | File size in bytes. |
| STATUS | <p>Indicates the status of the file on the agent computer. Possible values are:</p> <ul style="list-style-type: none"> • Exists This is the first time scanning for this file and it was found. • New This file was added to the file system of the agent computer since the last scan was performed. • Update This file exists in the new and previous scans. There have been changes to the date, time, size, and/or version. • Deleted This file was present in the previous scan but is missing in the new scan. • Not found No files were found that matched this request. |
| TIME | The time of the most recent modification to this file. |
| XTENSION | The file extension. This is useful for sorting and querying back-end database tables that store the data found in this object. |

WBEM Auditing

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



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

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

Figure 5 AUDIT.WBEM class instances.

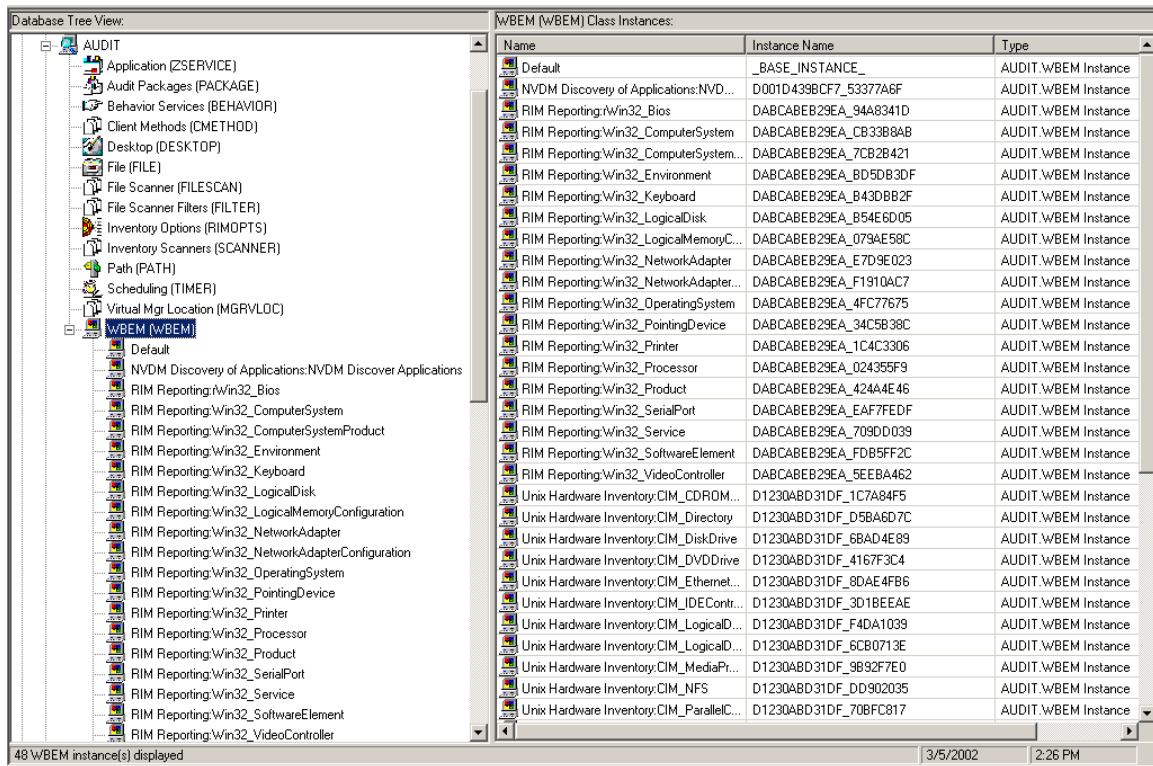


Table 15 below describes the attributes of the AUDIT.WBEM instance.

Table 15 AUDIT.WBEM Instance

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

| Attribute Name | Description |
|----------------|---|
| | Delete, Custom <ul style="list-style-type: none"> • Initial means that the file was found during the first scan of the agent computer. • New means that the file was found during the current scan. The file was not present during the previous scan. • Changed means that the file was present during the previous scan and is different from the file found during the current scan. • Deleted means that the file was found during the previous scan. The file is not present for the current scan. • Scan means that the file was found during the current scan. • Delete means that the file was found during the previous scan. The file is not present for the current scan. • Custom means that the file was found during a custom scan. The last three flags are not applicable to WBEM audits. |
| NAMESPACE | The name of the WBEM namespace to query or HARDWARE. |
| CLASS | The name of the WBEM class to query or HARDWARE. |
| PROPERTY | Specify one or more property names to be queried and reported. Use commas to separate more than one property name. If this attribute is blank, all properties in the class will be queried and reported. |
| CNDITION | An optional condition to narrow results of an audit. |
| OUTPUT | This is the name of the object to send to the Configuration Server. |

| Attribute Name | Description |
|----------------|--|
| TYPE | Indicates that WBEM scan is to be employed for this audit package. |
| NAME | Friendly name for this instance. This name will appear in the CSDB Editor's tree view to identify this instance. |



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

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

In addition to the attributes described in [Table 15](#) on page 84, the WBEM object also contains the following:

Table 16 WBEM Object Attributes in the Agent

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

| Attribute | Description |
|-----------|--|
| ZOBJTIME | The latest time under the current node. |
| ZRSCSRC | The name of the program promoted the resource. |

WBEM Object Processing

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

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

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

Disabling Remnant Configuration Server Instances for WBEM Object Processing

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

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

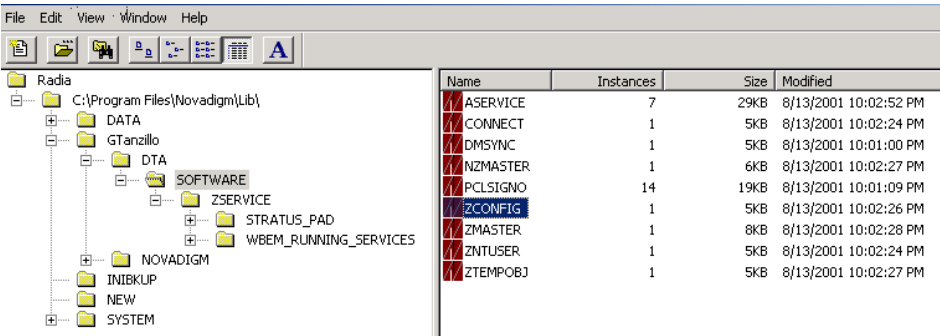
If these remnant instances exist or were imported into your CSDB, you must disable any configurations within them in order to ensure successful WBEM object processing.

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

Hardware Auditing

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

Figure 6 ZCONFIG object

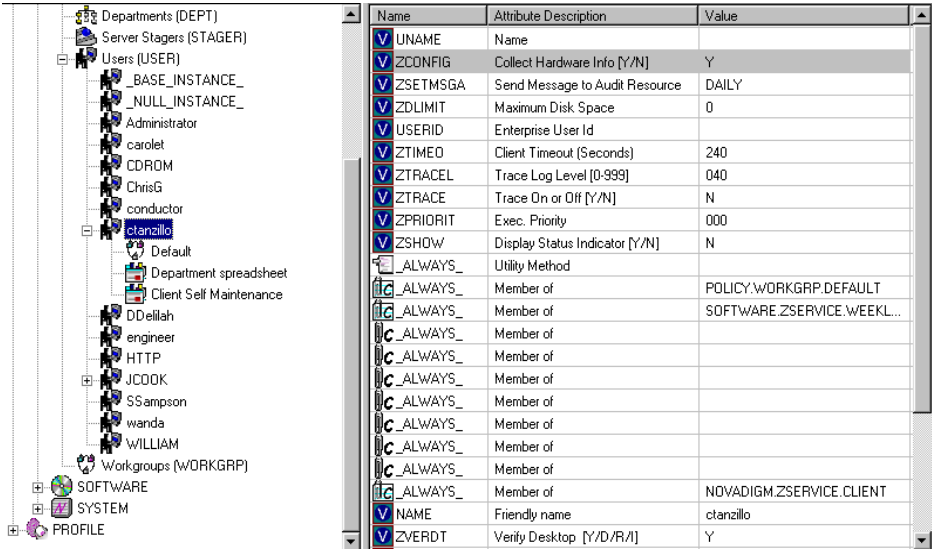


The screenshot shows a file explorer window with the address bar set to C:\Program Files\Novadigm\Lib\ and the left pane showing a tree structure of folders. The right pane displays a table of files and folders.

| Name | Instances | Size | Modified |
|----------|-----------|------|-----------------------|
| ASERVICE | 7 | 29KB | 8/13/2001 10:02:52 PM |
| CONNECT | 1 | 5KB | 8/13/2001 10:02:24 PM |
| DMSYNC | 1 | 5KB | 8/13/2001 10:01:00 PM |
| NZMASTER | 1 | 6KB | 8/13/2001 10:02:27 PM |
| PCLSIGNO | 14 | 19KB | 8/13/2001 10:01:09 PM |
| ZCONFIG | 1 | 5KB | 8/13/2001 10:02:26 PM |
| ZMASTER | 1 | 8KB | 8/13/2001 10:02:28 PM |
| ZNTUSER | 1 | 5KB | 8/13/2001 10:02:24 PM |
| ZTEMPOBJ | 1 | 5KB | 8/13/2001 10:02:27 PM |

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

Figure 7 POLICY.USER Class – ZCONFIG attribute



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

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

Figure 8 ZCONFIG object.

| Object | Variable | Heap | Options |
|--------|-----------|--------|--------------------------------------|
| 1 of 1 | | | |
| | Variable | Length | |
| | GATEWAY03 | 013 | 208.244.231.1 |
| | IPADDR01 | 007 | 0.0.0.0 |
| | IPADDR02 | 007 | 0.0.0.0 |
| | IPADDR03 | 015 | 208.244.231.104 |
| | LADAPT01 | 012 | 444553540000 |
| | LADAPT02 | 012 | 444553540001 |
| | LADAPT03 | 012 | 0050da644154 |
| | REBOOTD | 008 | 20010608 |
| | REBOOTT | 008 | 11:15:40 |
| | SUBNET01 | 007 | 0.0.0.0 |
| | SUBNET02 | 007 | 0.0.0.0 |
| | SUBNET03 | 013 | 255.255.255.0 |
| | ZGATEWAY | 011 | %GATEWAY03) |
| | ZHDWBIOS | 037 | 04/22/99 PhoenixBIOS 4.0 Release 6.0 |
| | ZHDWCDDR | 002 | E: |
| | ZHDWCOMP | 009 | ctanzillo |
| | ZHDWCPU | 007 | Pentium |
| | ZHDWCPUS | 006 | 450MHz |
| | ZHDWD00 | 002 | C: |
| | ZHDWD00C | 005 | Fixed |
| | ZHDWD00F | 014 | 10,109,403,136 |
| | ZHDWD00S | 005 | FAT32 |
| | ZHDWD00T | 014 | 13,689,888,768 |
| | ZHDWD01 | 002 | E: |
| | ZHDWD01C | 005 | CDROM |
| | ZHDWD02 | 002 | F: |
| | ZHDWD02C | 006 | Remote |
| | ZHDWD02F | 011 | 541,196,288 |
| | ZHDWD02S | 007 | NwCOMPA |
| | ZHDWD02T | 013 | 1,048,576,000 |
| | ZHDWD03 | 002 | G: |
| | ZHDWD03C | 006 | Remote |
| | ZHDWD03F | 014 | 51,560,579,072 |
| | ZHDWD03S | 004 | NTFS |

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

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

Table 17 ZCONFIG Object

| Attribute | Description |
|-----------|--------------------------|
| GATEWAY | Router for your subnet. |
| HALCOMP | Company of HAL.DLL |
| HALDATE | Date and time of HAL.DLL |

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

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

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

6 Successful Auditing

At the end of this chapter, you will:

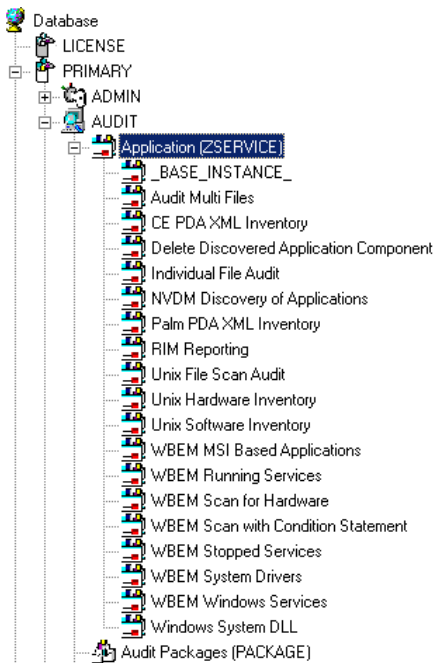
- Know how to use the pre-packaged Audit Applications (ZSERVICE).
- Know how to design your own Audit Packages (PACKAGE).

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

Sample Auditing

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

Figure 9 Sample Auditing services.



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

The sample audit services are described in the following table:

Table 18 Sample of Auditing Services

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

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

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

Configuring a Sample Audit

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

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

agent computer. The ACTION attribute indicates that the discovery of the file will be reported and sent to the Configuration Server for storage.

Figure 10 SCANFOR attribute of the Audit Multi Files instance.

Database Tree View

LICENSE

PRIMARY

ADMIN

AUDIT

Audit Application (ZSERVICE)

Audit Multi Files

Audit To Find and Capture MULTIPLE FILES

Audit To Find and Capture MULTIPLE FILES

Audit To Find and Capture MULTIPLE FILES

Differenced Audit on Install and Update

Audit Execute Behavior

Default

Delete Discovered Application Component

Individual File Audit

RIM Reporting

Unix Software Inventory Audit

WBEM MSI Based Applications

File class Audit To Find and Capture MULTIPLE FILES: Multi File Collection Instance

| Name | Attribute Description | Value |
|-----------|------------------------------------|-----------------------|
| SCANFOR | File name or pattern to search for | *\program files*.EXE |
| ACTION | Report Flags (I, N, C, D, S, D, C) | YYYYYYNN |
| OUTPUT | Output Object Name | FILEAUDT |
| TYPE | Scan Type (FILE) | FILE |
| GROUP | Group Name for audited component | |
| ZVERINFO | Collect Extended Info [1/0] | 1 |
| ZRSCSTYP | Server File Type (BINARY/TEXT) | BINARY |
| ZRSCMFIL | Manager Directory Location | |
| ZRSCVLOC | Virtual Manager Location | RADIA_UPLOAD |
| ZRSCMEM | PDS Member Name | |
| PRODUCT | Product Name | |
| PRODDVERS | Version | |
| ORGRNAME | Original File Name | |
| INTERNAL | Internal Name | |
| VERSION | Product Version | |

To configure a sample Audit package

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

HPCA Admin CSDB Editor - [1:CMCS - 1]

Database Tree View

LICENSE

PRIMARY

ADMIN

APPMGMT

AUDIT

Application (ZSERVICE)

BASE_INSTANCE_...

Audit Multi Files

CE POA XML Inventory

Delete Discovered Application Component

Individual File Audit

NVDM Discovery of Applications

Palm POA XML Inventory

RIM Reporting

UNIX File Audit

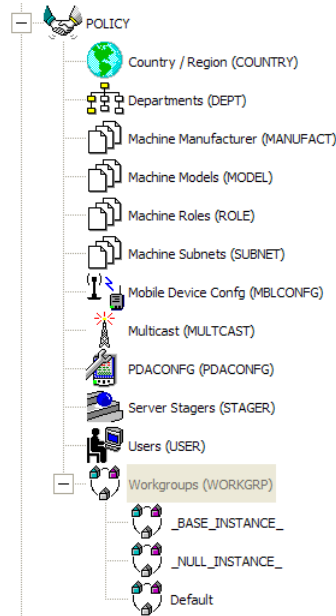
Application (ZSERVICE) Class Instances:

| Name | Instance Name | Type |
|--|--------------------------------|-------------------------|
| BASE_INSTANCE_... | BASE_INSTANCE_... | AUDIT.ZSERVICE Instance |
| Audit Multi Files | AUDIT_MULTI_FILES | AUDIT.ZSERVICE Instance |
| CE POA XML Inventory | CE_POA_XML_SVC | AUDIT.ZSERVICE Instance |
| Delete Discovered Application Component | DELETE_SAMPLE_COMPONENT | AUDIT.ZSERVICE Instance |
| Individual File Audit | INDIVIDUAL_FILE_AUDIT | AUDIT.ZSERVICE Instance |
| NVDM Discovery of Applications | NVDM_DISCOVERY_OF_APPLICATIONS | AUDIT.ZSERVICE Instance |
| Palm POA XML Inventory | PALM_POA_XML_SVC | AUDIT.ZSERVICE Instance |
| RIM Reporting | RIM_REPORTING | AUDIT.ZSERVICE Instance |
| UNIX File Audit | UNIX_FILE_AUDIT | AUDIT.ZSERVICE Instance |
| UNIX File Scan Behavior | UNIX_FILE_SCAN | AUDIT.ZSERVICE Instance |
| UNIX Hardware Inventory | UNIX_HARDWARE_INVENTORY | AUDIT.ZSERVICE Instance |
| UNIX Software Inventory | UNIX_SOFTWARE_INVENTORY | AUDIT.ZSERVICE Instance |
| UNIX User and Group Inventory | UNIX_USER_AND_GROUP_INVENTORY | AUDIT.ZSERVICE Instance |
| UNIX WBEM Computer System | UNIX_WBEM_COMPUTERSYSTEM | AUDIT.ZSERVICE Instance |
| UNIX WBEM Operating System | UNIX_WBEM_OPERATINGSYSTEM | AUDIT.ZSERVICE Instance |
| UNIX WBEM Process | UNIX_WBEM_PROCESS | AUDIT.ZSERVICE Instance |
| UNIX MSI Based Applications | MSI_INSTALLED_SOFTWARE | AUDIT.ZSERVICE Instance |
| WBEM Running Services | WBEM_RUNNING_SERVICES | AUDIT.ZSERVICE Instance |
| WBEM Scan for Hardware | WBEM_SYSTEM_HARDWARE_AUDIT | AUDIT.ZSERVICE Instance |
| WBEM Scan for User Account and Group Information | WBEM_USER_GROUP | AUDIT.ZSERVICE Instance |
| WBEM Scan with Condition Statement | WBEM_CONDITION | AUDIT.ZSERVICE Instance |
| WBEM Stopped Services | WBEM_STOPPED_SERVICES | AUDIT.ZSERVICE Instance |
| WBEM System Drivers | WBEM_SYSTEM_DRIVERS | AUDIT.ZSERVICE Instance |
| WBEM Windows Services | WINS3_SERVICE_AUDIT | AUDIT.ZSERVICE Instance |
| Windows System DLL | AUDIT_SYSTEM_DLL | AUDIT.ZSERVICE Instance |

- 4 Scroll to and expand the POLICY Domain.

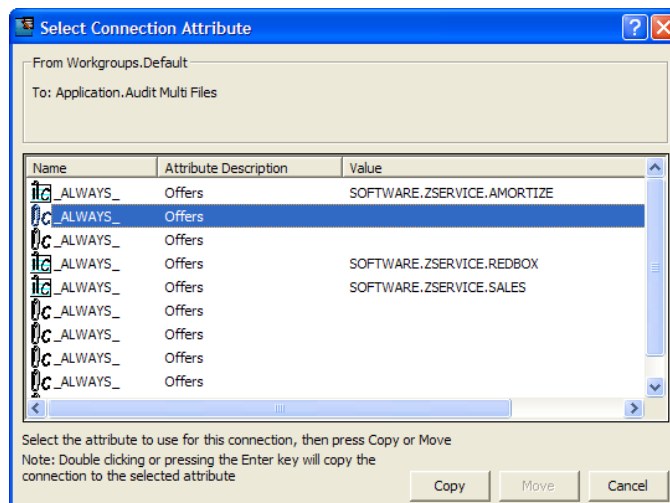
For our example, we would like all users that are members of the Workgroup class to select this audit package from their Application Self-service Manager.

5 Expand the POLICY.WORKGROUPS class.

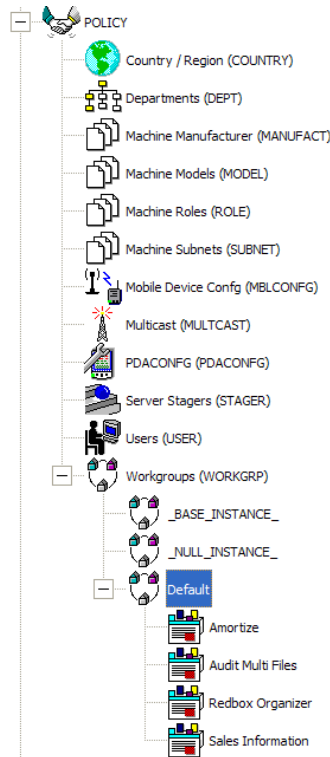


6 Select the **Audit Multi Files** package from the **ZSERVICE** class and drag it to the POLICY.WORKGROUPS class and drop it on the **Default** instance.

The Select Connections Attribute window opens.



- 7 Click **Copy** to add this package. The Confirm Connection dialog box opens.
- 8 Click **Yes** to confirm the connection. The Audit Multi Files package is added to WORKGRP Class.




The collection of inventory information occurs on the Inventory Manager agent computer when a user connects to the Configuration Server as follows:

- Through an Application Self-service Manager agent connect, when the user launches that program.
- or
- Through the Application Manager agent when the user double-clicks the Connect icon on his desktop, or is scheduled or notified to connect.

Figure 11 on page 101 shows the available Audit Multi Files package that an Application Self-service Manager user would see when connecting to the Configuration Server:

Figure 11 Application Self-service Manager shows Audit Multi Files

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

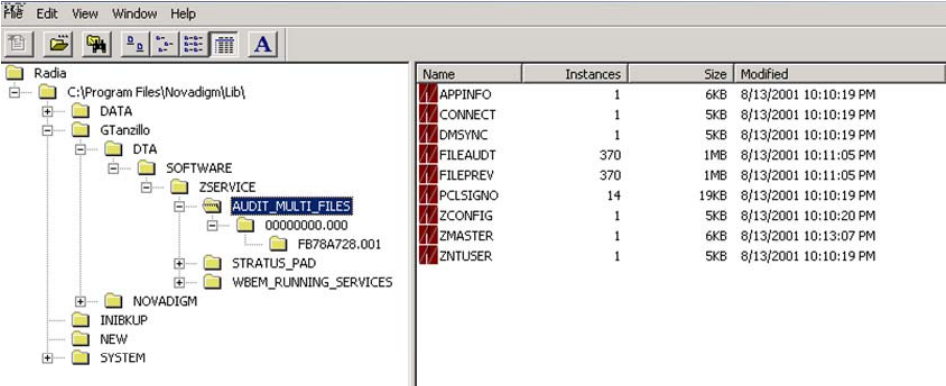
When the subscriber selects and installs the Audit Multi Files package from the Application Self-service Manager, there are really two connections. The first connection downloads the Audit service. The second connection sends the audit results back to the Configuration Server. The audit-related scans are done between the two connections.

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

What Happened during the Scan?

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

Figure 12 AUDIT_MULTI_FILES in the LIB directory



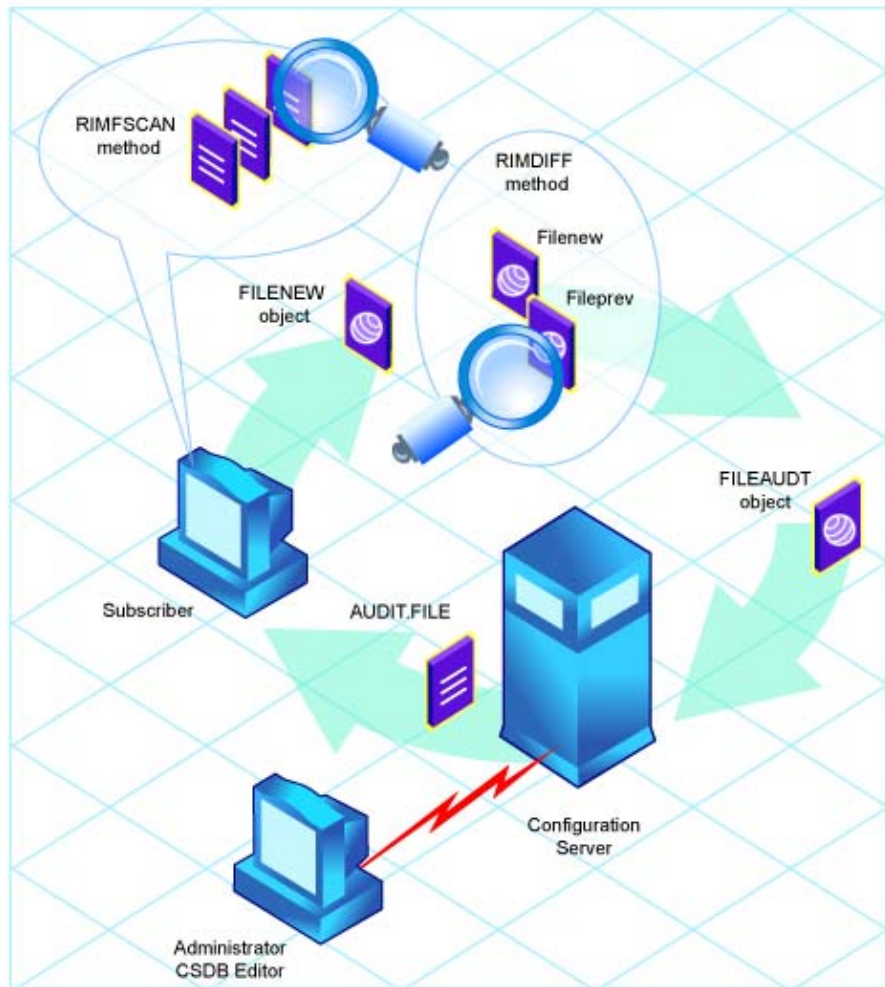
| Name | Instances | Size | Modified |
|----------|-----------|------|-----------------------|
| APPINFO | 1 | 6KB | 8/13/2001 10:10:19 PM |
| CONNECT | 1 | 5KB | 8/13/2001 10:10:19 PM |
| DMSYNC | 1 | 5KB | 8/13/2001 10:10:19 PM |
| FILEAUDT | 370 | 1MB | 8/13/2001 10:11:05 PM |
| FILEPREV | 370 | 1MB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

Within the ZSERVICE, note the two objects, **FILEAUDT** and **FILEPREV**. These objects are created and stored in the ZSERVICE of the LIB directory

whenever an audit package is installed. The FILEAUDT object contains one heap for each file discovered during the auditing scan. It also contains the attributes from the AUDIT.FILE instance that controlled the scan.

The AUDIT.FILE class instances in an audit package control the auditing for files on the agent computer. The RIMFSCAN and the RIMDIFF methods on the agent computer perform the actual file auditing operations by specifying what files to look for.

Figure 13 Auditing with the RIMFSCAN and RIMDIFF methods



- The RIMFSCAN method scans the Client Automation agent's file system based on the values in the AUDIT.FILE class instance in the audit package. It constructs an object called FILENEW. The FILENEW object contains one heap per file discovered during the current scan.
- The RIMDIFF method compares scan results from the current scan (the scan done during the current agent connect stored in the FILENEW object) with scan results from a previous scan (the scan done during a previous agent connect process stored in the FILEPREV object). It will construct the FILEAUDT object that is then sent to the Configuration Server. The RIMDIFF method then deletes the FILEPREV object and will rename the FILENEW object to FILEPREV.

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

7 Creating Audit Packages

At the end of this chapter, you will:

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

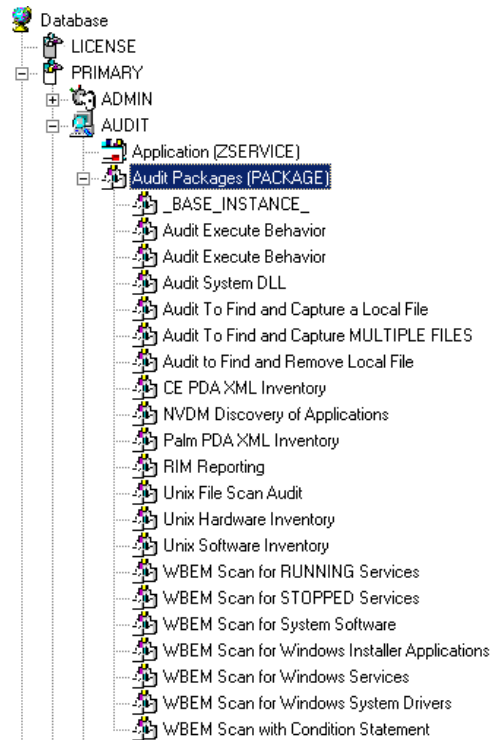
This manual is provided to assist you with installing and implementing the Inventory Manager. Choose the appropriate strategies suited for your enterprise needs.

Audit Packages or PACKAGE Class

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

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

Figure 14 Audit Package (PACKAGE) class.




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

Figure 15 Individual File Audit instance.



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

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

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

Table 19 Inventory Options (RIMOPTS) Class

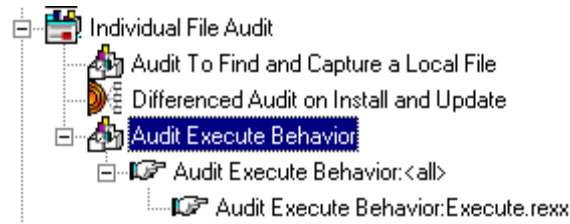
| Instance | Description |
|----------|---|
| Default | <p>Contains the base instance attributes for the RIMOPTS Class.</p> <ul style="list-style-type: none">• Collect attribute is set to Diff.• Runexec attribute is set to IU.• ZSVCTYPE attribute is set to I. |

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

for additional information about RIMOPTS attributes.

Finally, a connection to an auditing behavior is needed.

Figure 16 Connection to an Audit Behavior



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

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

Using the CSDB Editor Create/Maintain Audit Services

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

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

- What am I auditing for? Will it be a hardware audit, a file audit, or a WBEM object audit?
- Will I be deploying to all users, or a select few?
- Will I want this to be connected to a timer for scheduled deployment? (See [Configuring Timers for Audit Collection](#) on page 121 for information concerning timers.)

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

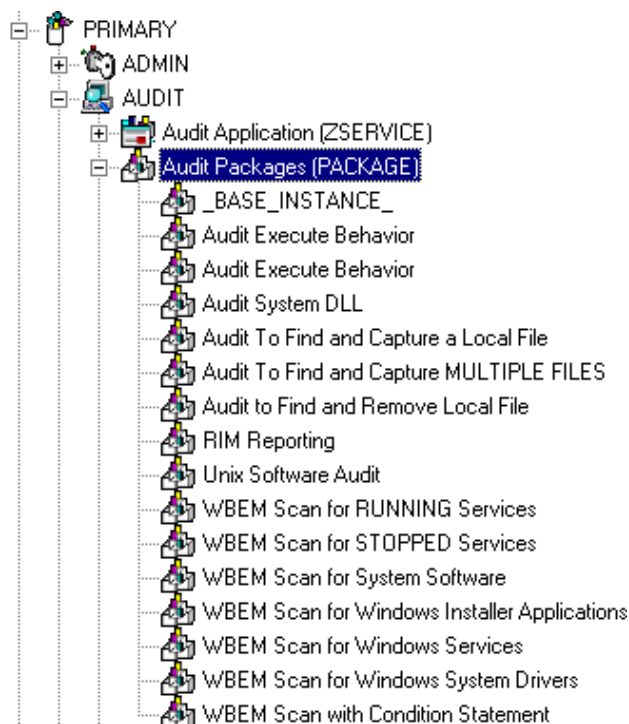


If you are creating a WBEM Audit Package, be aware Inventory Manager leverages Microsoft's Windows Management Instrumentation (WMI) to collect hardware and software inventory data by using WMI queries. Some WMI queries can traverse the network contacting other servers in the enterprise to collect the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

To create a new Audit package

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

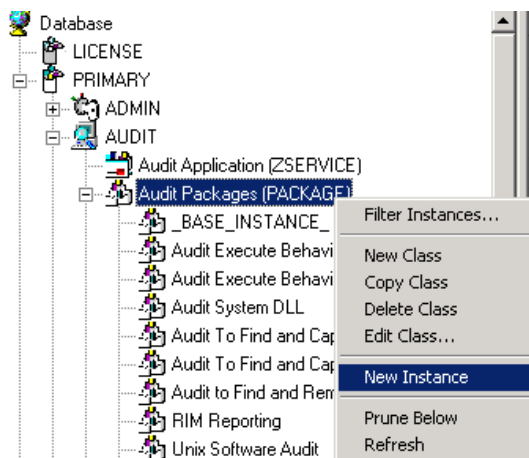
The User ID, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. Check with your security administrator to obtain your own User ID and Password, if necessary.
- 3 Double-click **PRIMARY**.
- 4 Expand the **AUDIT Domain**.
- 5 Double-click on **Audit Packages (PACKAGE)** class.



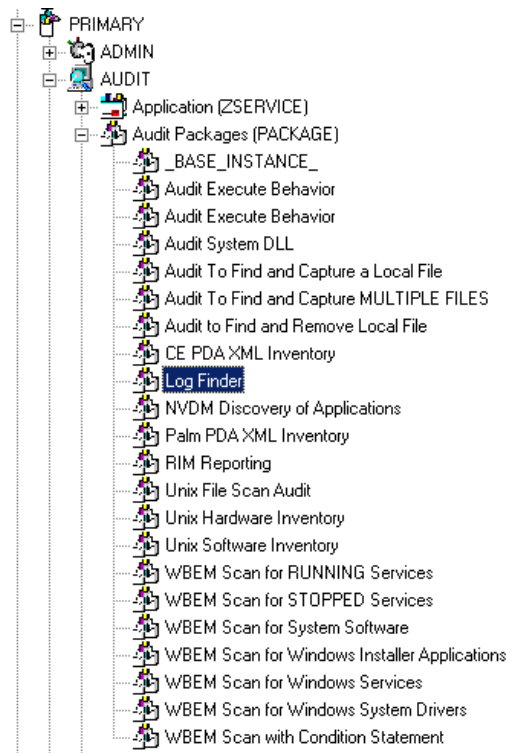
As an example, we will create a new auditing package called Log Finder. This package will scan a user's computer for .log files, capture them, and return the results to the administrator.

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

A shortcut menu opens.



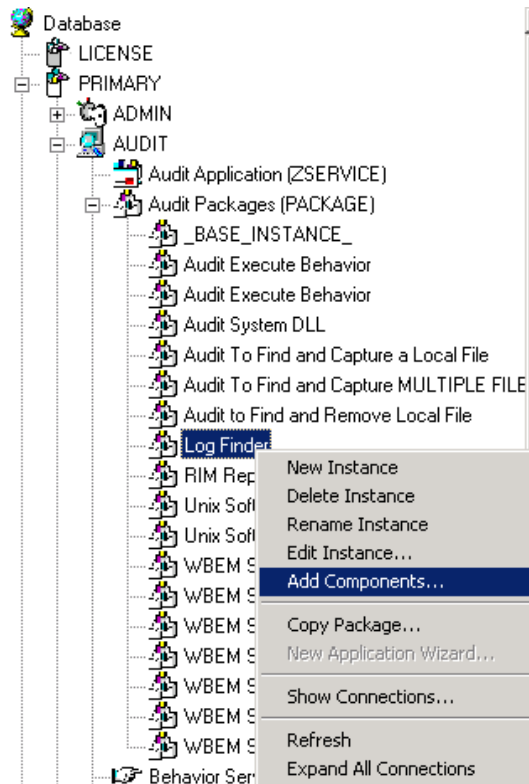
- 7 Select **New Instance** from the shortcut menu. The Create Instance dialog box opens.
- 8 Enter a new display name for the package instance. This friendly name will appear in the tree view.
- 9 Enter a name for the Create a new Audit Packages (PACKAGE) instance name. This name appears in the title bar of the list view of the CSDB Editor window when the instance is selected and opened in the tree view.
- 10 Click **OK** to continue. The new Log Finder package is added to the AUDIT.PACKAGE Class.



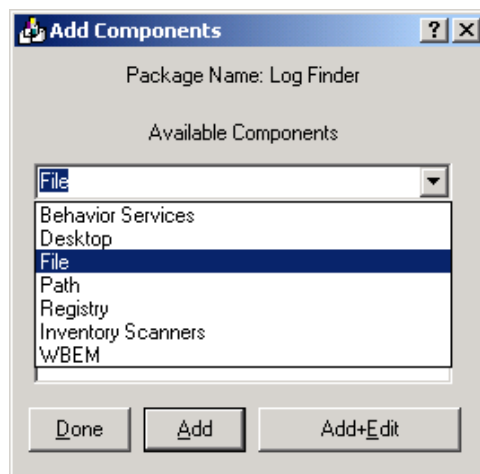
Once the Log Finder package is created, you will need to add its components.

To add a component to an Audit package

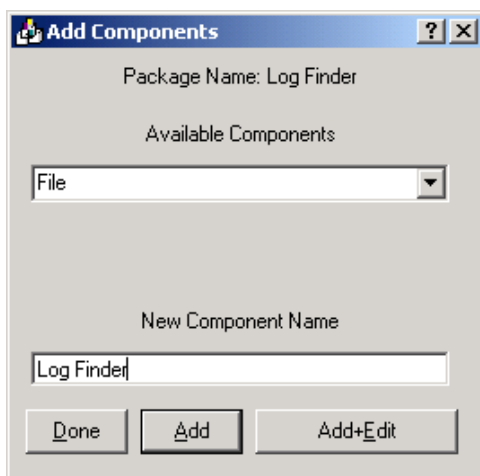
- 1 Right-click on the Log Finder package.
- 2 A shortcut menu opens.



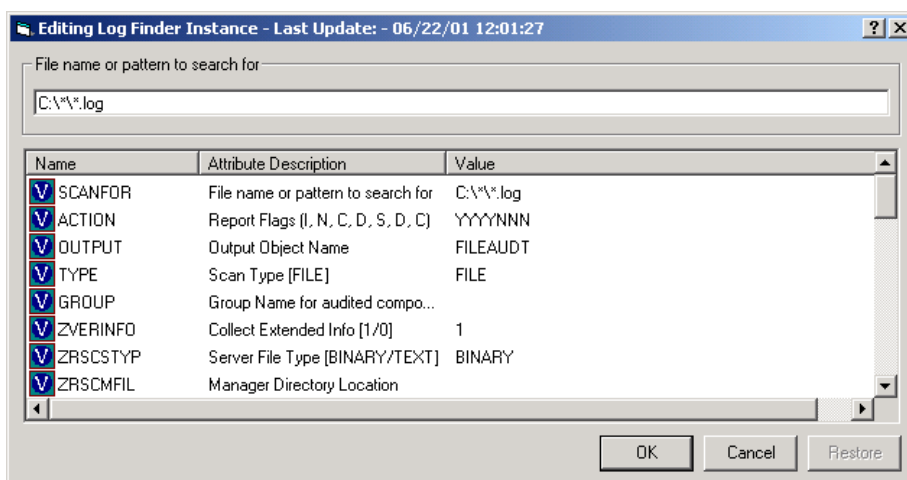
- 3 Select **Add Component** from the shortcut menu. The Add Components dialog box opens.
- 4 Click on the **Available Components** drop-down arrow, and select **File**.



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



- 6 Click **Add+Edit**. The component is added to the package and the Editing Instance dialog box opens.



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



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

For our example, we changed the SCANFOR attribute to C:**.log. Continue to edit, line-by-line, as necessary.

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

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

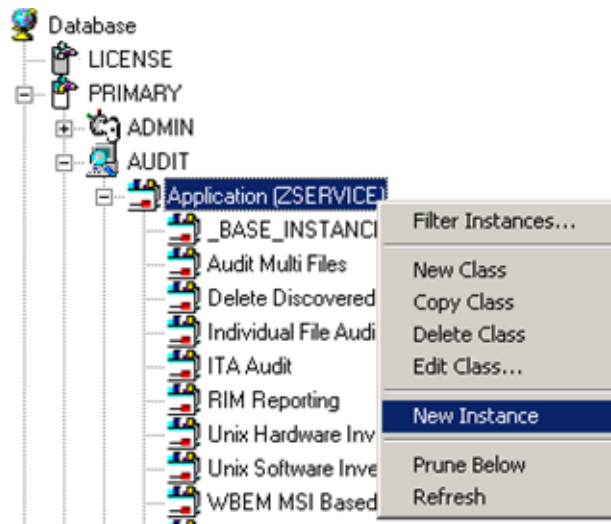
Next, you will need to create a ZSERVICE instance to contain the Log Finder package.

To create a ZSERVICE instance

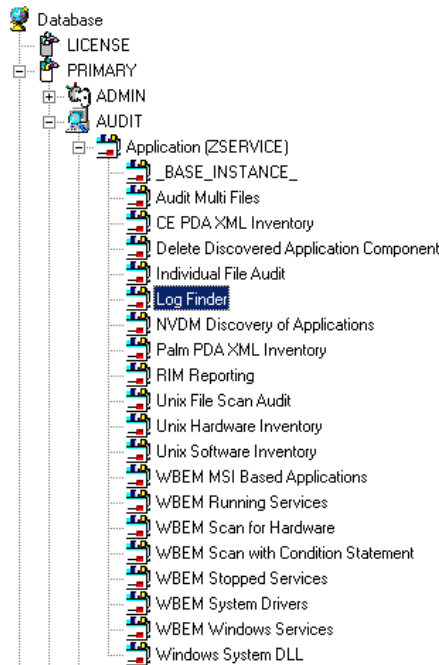


While working in the AUDIT Domain, note that the New Application Wizard is *not* available to connect a package to a service. You need to either copy an existing instance or create a new one.

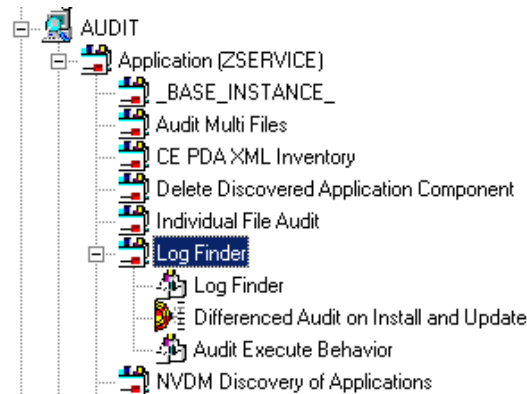
- 1 In the CSDB Editor, expand the AUDIT.ZSERVICE class in the tree view.
- 2 Right-click **Audit Application (ZSERVICE)** and a shortcut menu opens.



- 3 Select **New Instance** from the shortcut menu. The Create Instance dialog box opens.
- 4 Type a display and an instance name.
- 5 Click **OK**. The ZSERVICE Log Finder is added to the AUDIT.ZSERVICE class.



- 6 Use the CSDB Editor to connect the Log Finder package to the Log Finder service.



Once the connection to the ZSERVICE has been completed, various optional steps can be taken.

You might want to ask yourself the following questions:

































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

- Will this be a mandatory or optional service?
- Will the service have a certain length of time to be active?
- Do I want to confirm if the service is installed or not?

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

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


Figure 17 Log Finder ZSERVICE attributes

| Application class: Log Finder Instance Attributes: | | |
|--|-------------------------------------|-----------------------------------|
| Name | Attribute Description | Value |
|  ZSTOP000 | Expression Resolution Method | |
|  ZSTOP001 | Expression Resolution Method - 001 | |
|  ZSTOP002 | Expression Resolution Method - 002 | |
|  ZSTOP999 | Stop Unless Radia Connect | |
|  ZSVCNAME | Service Name/Description | Log Finder |
|  ZSVCTTYP | Application Target Type [A/S] | |
|  ZSVCMD | Mandatory or Optional Service [M/O] | M0 |
|  ZSVCCSTA | Service Status on Client (999) | 999 |
|  ZSVCPRI | Service Create Ordering [01-99] | |
|  _ALWAYS_ | Contains | AUDIT.PACKAGE.LOG_FINDER |
|  _ALWAYS_ | Contains | |
|  _ALWAYS_ | Contains | |
|  _ALWAYS_ | Contains | |
|  _ALWAYS_ | Contains | |
|  _ALWAYS_ | Contains | AUDIT.RIMOPTS.DIFF_INSTALL_UPD... |
|  _ALWAYS_ | Contains | AUDIT.PACKAGE.AUDIT_EXECUTE_B... |
|  _ALWAYS_ | Utility Resolution Method | |
|  ZCREATE | Service Installation Method | |
|  ZINIT | Service Initialization Method | |
|  ZDELETE | Service Delete Method | |
|  ZUPDATE | Service Update Method | |
|  ZVERIFY | Service Verify Method | |
|  ZREPAIR | Service Repair Method | |
|  ZAVIS | Available,Verified,Installed,Sync F | YXNX |
|  PUBDATE | Published Date of Service | |
|  VERDATE | Verified Date of Service | |
|  UPGDATE | When Application was Upgraded on De | |
|  UPDDATE | Upgrade Date (Programmatic) | |
|  INSTDATE | Installed Date | |
|  DELDATE | Delete Date | |
|  AUTHOR | Author Name | |
|  DESCRIPT | Application Description | |

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

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

Figure 18 Log Finder in the Application Self-service Manager

| Name | Status | Compressed Size | Description | Mandatory | |
|-------------------|-----------|-----------------|-------------|-----------|---|
| Amortize | Available | n/a | | O | |
| Drag & View | Available | 2.51 MB | | O | |
| GS-CALC | Available | n/a | | O | |
| Log Finder | | | | | Size 6.67 KB |
| | | | | | Compressed Size 2.34 KB |
| Available | | | | |  |
| Redbox Organizer | Available | n/a | | O | |
| Sales Information | Available | n/a | | O | |
| StratusPad | Available | n/a | | O | |

8 Configuring Timers for Audit Collection

At the end of this chapter, you will:

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

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

The Scheduling (TIMER) Class

The Scheduling (TIMER) class enables the Client Automation administrator to set a timer on the Client Automation agent computer that will cause one or more audit services to be processed whenever the timer expires. The administrator can use this method to process mandatory audit services automatically according to a pre-determined schedule.

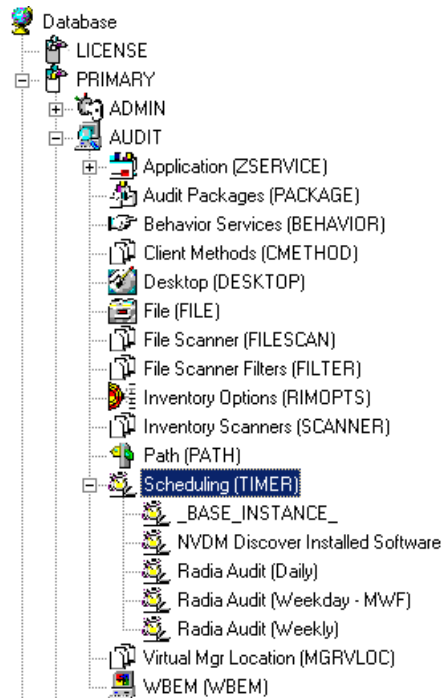
▶ As distributed by HP, the SOFTWARE Domain also contains a Scheduling (TIMER) class. Timers can be specified in instances of either of these Scheduling (TIMER) classes and can be connected to an Application (ZSERVICE) class instance in either the SOFTWARE or AUDIT Domain.

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

- **Daily**
which will deploy a ZSERVICE everyday at the time specified.
- **Weekday**
which will deploy a ZSERVICE on Mondays, Wednesdays, and Fridays at a specified time.
- **Weekly**
which will deploy a ZSERVICE every seven days at a specified time.
- **Discover Installed Software Timer**
executes a ZSERVICE weekly between 8:30 am and 10:30 pm. Use this particular timer in conjunction with the ZSERVICE Discovery of Applications that audits the ADD/REMOVE PROGRAM part of the OS.

These sample packages can be copied and modified, changing the time parameters to suit your needs. Refer to the *Administrator Guide* for information on copying an instance. Or, you can create a new timer instance by following the instructions given in [Creating a Timer Instance](#) on page 127.

Figure 19 AUDIT Scheduling (TIMER) class



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

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

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

Table 20 Scheduling (TIMER) Class

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

| Attribute | Description |
|-----------|---|
| | <ul style="list-style-type: none"> • Before 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00. • After 4:00:00, the command in the instance will be executed the <i>next</i> day at 4:00:00. <p>Example 2: Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,04:00:00) If ZSCHTYPE = IMMEDIATE and it is:</p> <ul style="list-style-type: none"> • Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00. • Friday and after 4:00:00, the command in the instance will be executed immediately. <p>If ZSCHTYPE = DEFERRED and it is:</p> <ul style="list-style-type: none"> • Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00. • Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00. |
| ZSCHFREQ | This attribute indicates how often the timer should expire according to the frequency specified in the ZSCHDEF attribute. <ul style="list-style-type: none"> • Once for a one-time expiration. • Periodic for a repeated expiration. • Random for random intervals. |
| ZRSCCMDL | This attribute indicates the command line that is executed on the subscriber's computer when the timer expires. |
| ZSVCOID | Specifies the object ID of the Application instance that this Scheduling instance is connected to. This value is inherited from the base instance and should not be modified. |
| _ALWAYS_ | Stores the connections to other instances. |
| NAME | Friendly name for this instance. |

| Attribute | Description |
|------------------|---|
| APPSVC | Application. |
| REQUEST | Application request. |
| DOMAIN | Server's domain name. |
| IPADDR | Server's IP address/name. |
| SOCKET | Server's socket number. |
| MGRNAME | Server's name. |
| ZCREATE | Scheduler CREATE method that runs on the Client Automation agent computer. This value is inherited from the base instance and should not be changed. |
| ZVERIFY | Scheduler VERIFY method that runs on the Client Automation agent computer. This value is inherited from the base instance and should not be changed. |
| ZUPDATE | Scheduler UPDATE method that runs on the agent computer. This value is inherited from the base instance and should not be changed. |
| ZDELETE | Scheduler DELETE method that runs on the Client Automation agent computer. This value is inherited from the base instance and should not be changed. |
| RUNSYNC | Sets the value of Yes or No for the synchronous timer execution. The default value is Yes. |
| ZNOPING | Controls the automatic sensing of a network connection between the Client Automation agent computer and the Configuration Server. An expired time will continually evaluate whether communications with the Configuration Server can be established. When communications are established, the command line associated with the time is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again. |


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

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

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

Creating a Timer Instance

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

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

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

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

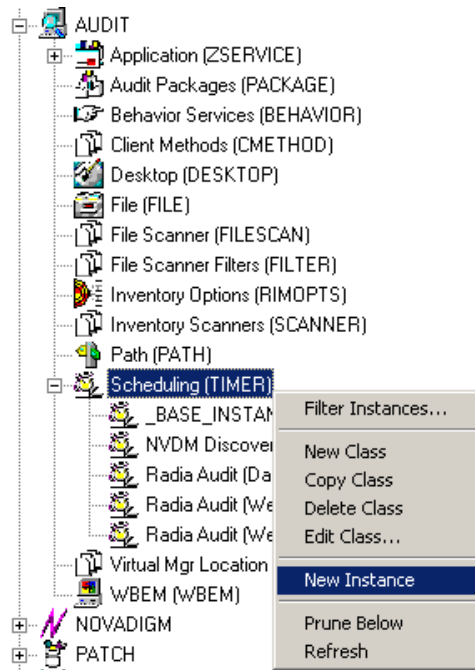
To create a new timer in the AUDIT Domain

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

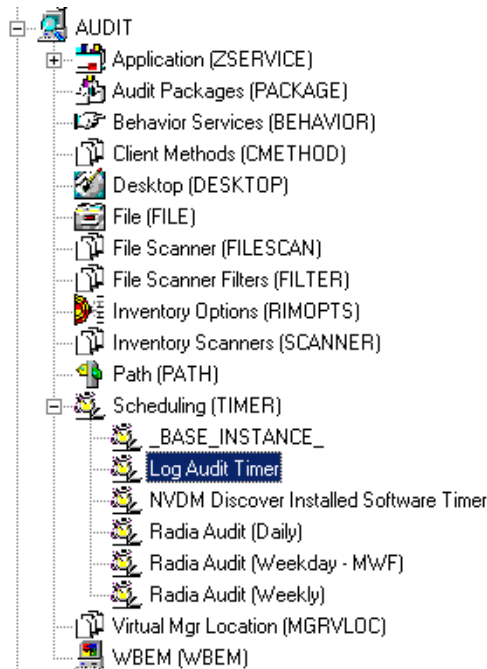


The User ID, as shipped from HP, is **RAD_MAST.**, with no password necessary. This may have been changed during installation. Check with your security administrator to obtain your own User ID and Password, if necessary.

- 3 Double-click **PRIMARY**.
- 4 Double-click **AUDIT**.
- 5 Right-click **Scheduling (TIMER)**.



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



Specifying Timer Settings


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

Specifying ZSCHDEF

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

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

WEEKDAY(<DAY of Week>,<TIME>[,<LIMIT>])
NUMDAYS(<DATE>,<TIME>[,<LIMIT>],<Number of Days>)
INTERVAL(<DATE>,<TIME>[,<LIMIT>],<Number of Seconds>)

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

NUMDAYS: NUMDAYS(20000803,08:00:00,12:00:00,14)
NUMDAYS: NUMDAYS(20000803,08:00:00,,14)

- The value of freq can be:
DAILY, WEEKLY, WEEKDAY, HOURLY, INTERVAL, NUMDAYS
- If the value of freq is DAILY, WEEKLY, HOURLY, INTERVAL, or NUMDAYS, the date is then specified in the following form:
YYYYMMDD
- If the value of freq is WEEKDAY, the date is then specified as the name of a day of the week in all uppercase letters. This would be one of the following:
MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY
- The values for time and limit_time are optional. They are specified in the following form:
HH:MM:SS
- The value for count is optional. It is specified as an integer.

The timer expiration can also be configured on the value of ZSCHFREQ. Use [Table 20](#) on page 124 to help you determine the appropriate syntax.

Table 21 Syntax of ZSCHDEF Attributes

| Type | Syntax | Timer Expires |
|---------|------------------------------------|---|
| DAILY | DAILY(&ZSYSDATE,24:00:00) | Daily at midnight by the system's date. |
| WEEKLY | WEEKLY(&ZSYSDATE,01:00:00) | Every seven days at 1:00 am. |
| WEEKDAY | WEEKDAY(Name of Weekday*,01:00:00) | Every <i>Name of Weekday*</i> at 1:00 AM. The weekday must be specified in uppercase. |

| Type | Syntax | Timer Expires |
|----------|-----------------------------------|--|
| HOURLY | HOURLY (&ZSYSDATE,08:41:00) | Hourly starting at 8:41 AM on the system's date. |
| INTERVAL | INTERVAL (&ZSYSDATE,08:41:00,,30) | Every 30 minutes starting at 8:41 AM based on system's date. |
| NUMDAYS | NUMDAYS (20000803,08:00:00,,14) | Every 14 days starting on August 3, 2000 at 8:00 AM. |

Specifying ZSCHTYPE

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

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

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

Specifying ZSCHFREQ

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

Specifying ZRSCCMDL

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

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

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



Execution causes Client Automation to launch the AUDIT service behavior, (EXECUTE.REXX) attached to the AUDIT service.

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

Specifying ZNOPING, PINGDLAY, and PINGCNT

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

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

- If the ZNOPING attribute is not in the ZTIMEQ object, or if ZNOPING is not equal to N, the Scheduler service does not ping the Configuration Server.
- If ZNOPING = N, the Scheduler service will ping the Configuration Server.
 - If the Configuration Server is pinged successfully, the command in the ZRSCCMDL attribute is executed. The PENDING attribute in the Client Automation agent's ZTIMEQ object is then set to N. This indicates that the Scheduler service does not need to ping the Configuration Server again.
 - Set ZNOPING to **W** if you are specifying an end limit in the ZCHDEF attribute. The Scheduler pings the Configuration Server before executing the command. If the Configuration Server is unavailable, the ZPENDING flag is set to "W". If the ZSCHEDEF has a limit time, then when that time passes, the ZPENDING flag is set to N, and the Scheduler will not attempt to execute the command until its next scheduled time.
 - If the Configuration Server is not pinged successfully, the timer is not processed any further. The ZPENDING attribute value remains set to Y. The next time the Scheduler service expires, it should ping the Configuration Server again.

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

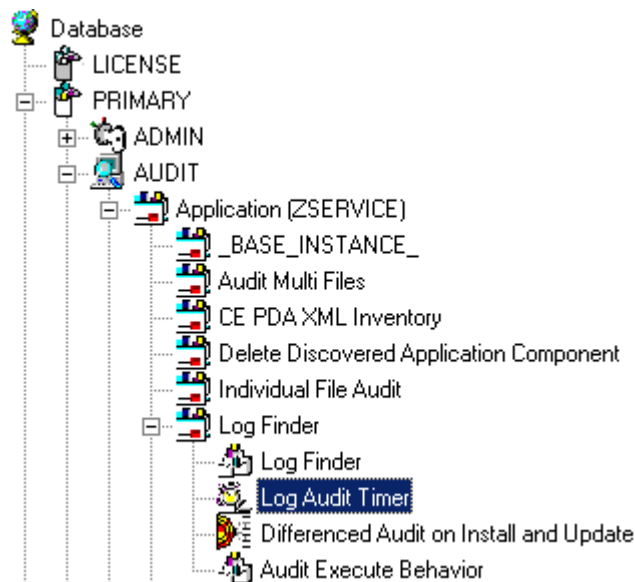
- If ZNOPING is set to N, PINGDLAY specifies the time in milliseconds between pings. The default is 2000.
- If ZNOPING is set to N, PINGCNT specifies number of ping attempts. The default is 3 attempts.

Connecting the Timer to a Service

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

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

Figure 20 Log Audit Timer instance connected to Log Finder service



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

Figure 21 Log Finder attached to a user



Audit Execution Configuration

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

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

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

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

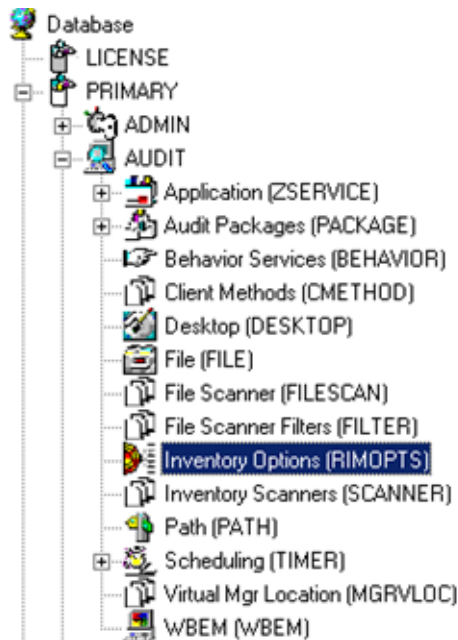
To customize the RIMOPTS instance

- 1 From the Start menu, select **Programs → HP Client Automation Administrator → HP Client Automation Administrator CSDB Editor**. The CSDB Editor Security Information dialog box opens.



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

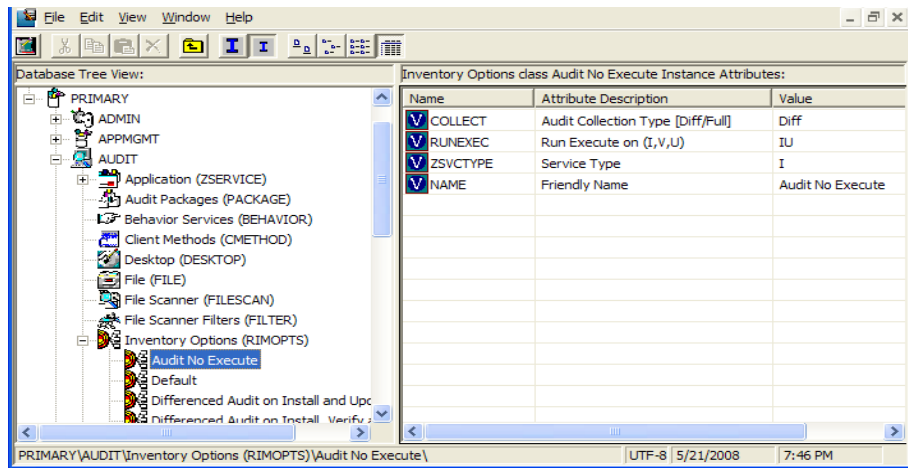
- 2 If necessary, type a User ID and Password, and then click **OK**. The CSDB Editor window opens.
- 3 Expand the **PRIMARY File** and the **AUDIT Domain**.



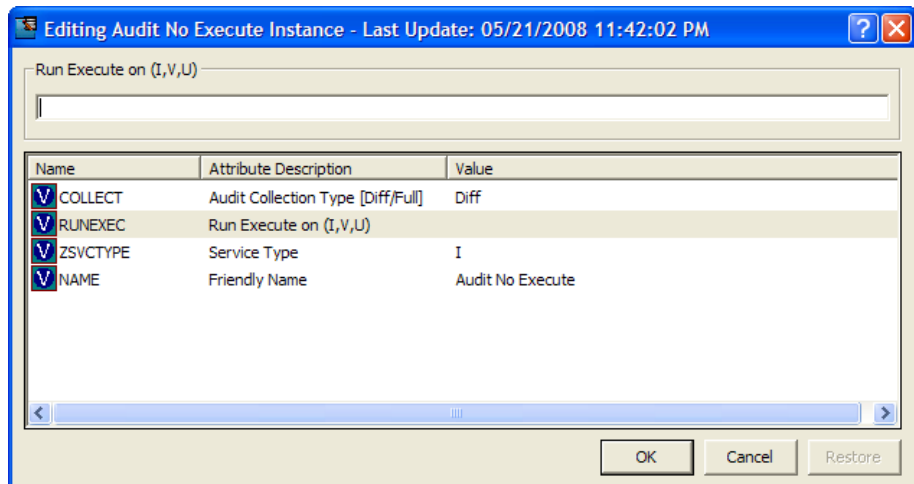
- 4 Create a new instance in the **Inventory Options (RIMOPTS)** class called CM_AUDIT_NO_EXECUTE, and click **OK**. The Create Instance dialog box opens.

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

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

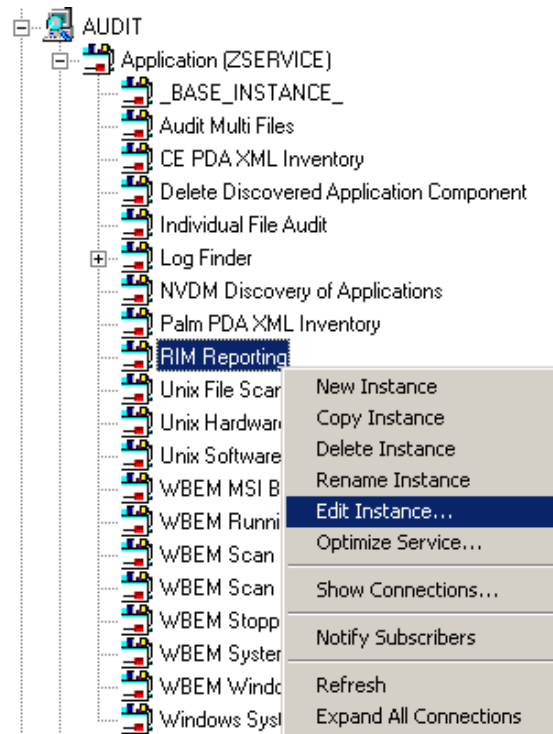


- 6 Double-click the **RUNEXEC** attribute in the list view to edit it. Remove any attribute information. This will ensure that the audit service will not run during the installation, verification, or update function.

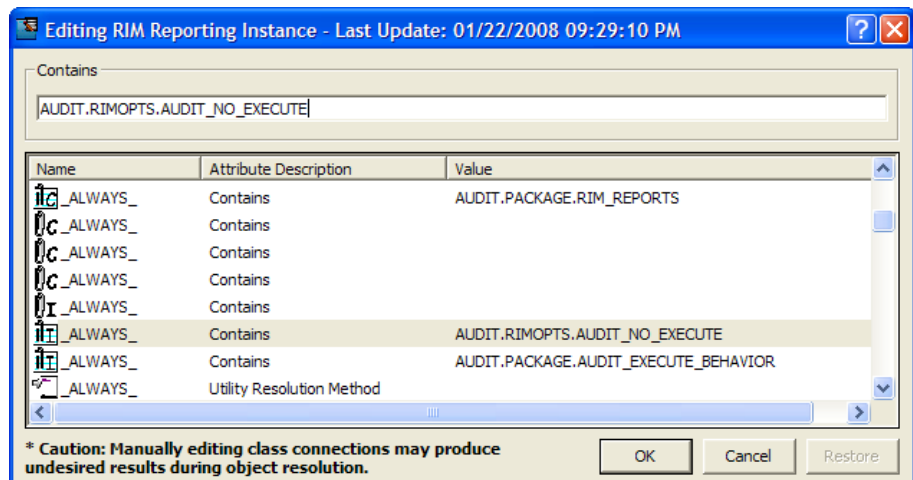


Next, determine which AUDIT service you will be adding the new RIMOPTS service to. For example, select the RIM_REPORTING service.

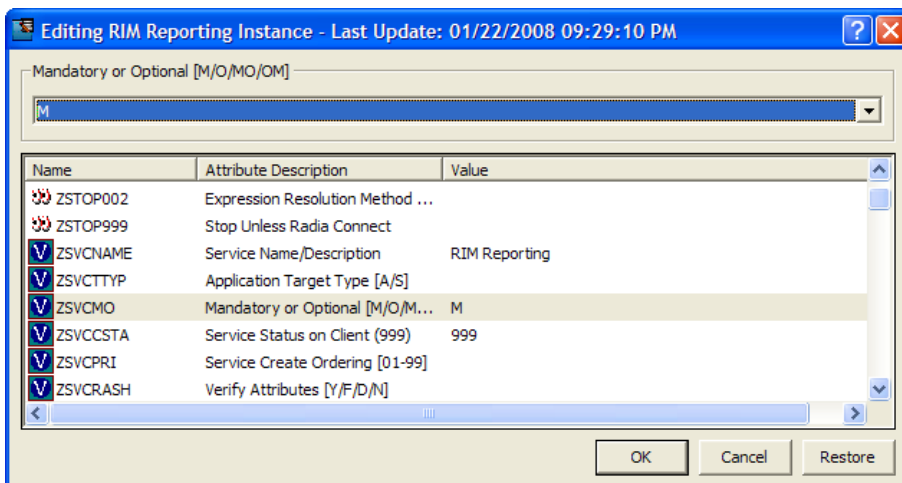
- 7 Right-click on **RIM_REPORTING** Service in the AUDIT class.
- 8 Select **Edit Instance**.



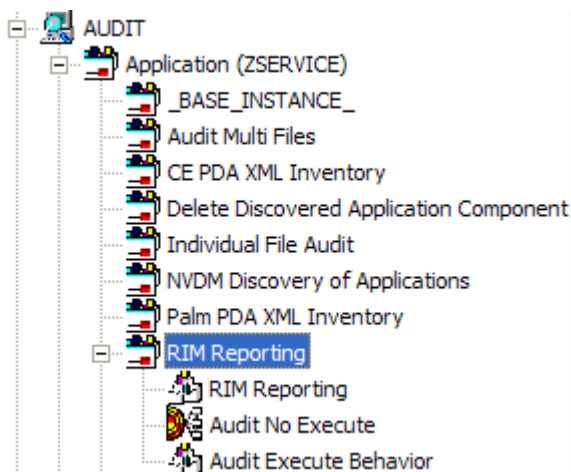
- 9 Locate the `_ALWAYS_Contains` attribute with the value of `AUDIT.RIMOPTS.DIFF_INSTALL_UPDATE` and change it to a value of `AUDIT.RIMOPTS.AUDIT_NO_EXECUTE`.



- 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 Client Automation agent.



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



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

9 Viewing Inventory from the Reporting Server

At the end of this chapter, you will:

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

Accessing the Reporting Server

To access the Reporting Server

- Open a Web browser and type the following address:

`http://<hostname>/reportingserver`

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



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

Viewing Audit Information Using the Reporting Server

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



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

Reporting Views for Inventory Reports

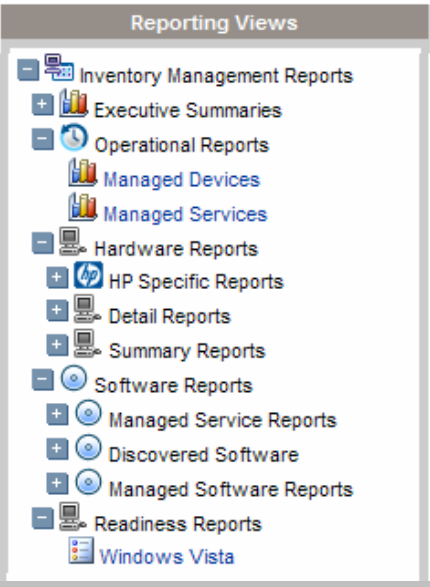
To view the reports, first access your Reporting Server. Then, under Reporting Views, click **Inventory Management Reports** to expand the list of reports.

There are different types of Inventory Management Reports:

- Executive Summaries
- Operational Reports
- Hardware Reports

- Software Reports
- Readiness Reports

Figure 22 Inventory Management reports



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

Table 22 Hardware Reporting Views

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

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

Table 23 Software Reporting Views

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

Windows Vista Readiness Reports

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

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

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

To display Windows Vista Readiness reports

- 1 In the Display Options area, select **Inventory Management Reports**.
- 2 Select **Readiness Reports**.
- 3 Select Windows Vista.
- 4 View the reports and charts available to determine the Windows Vistas upgrade readiness of your devices. The Readiness Status and Additional Information columns contain information about the current level of readiness for each device.

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

Filtering Inventory Reports with Reporting Server

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

Filter types include:

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

Figure 23 Inventory Management Related Data Filters

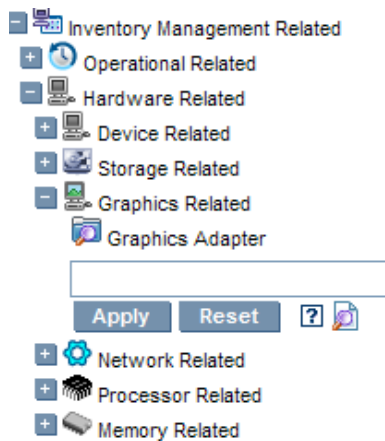


Expand each individual Inventory Management Related Data Filter to see the available filters you can apply to the current Reporting View.

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

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

Figure 24 Expand a filter



Click the magnifying glass to open the filter lookup window.

Figure 25 **Select the filter.**

Graphics Adapter

Manual Input

☐ %

All Records Retrieved from Database

☐ Intel(R) 82852/82855 GM/GME Graphics Controller

☐ RAGE XL PCI

☐ VMware SVGA II

☐ Winvnc video hook driver

Click any of the available criteria check boxes to select the criteria you would like to use in your filter. For additional information on creating filters and using the Reporting Server in general, refer to the *Reporting Server Guide*.

A Detail and Summary Reporting Tables

Table 24 Inventory Reporting – Detailed Reports

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

| Action | Displayed Table Title | Columns Queried /Display Name | Tables Queried |
|-----------------------------------|-----------------------|--|------------------------------|
| | | wName/Name | |
| | | wVendor/Vendor | |
| | | wVersion/Version | |
| | | wInstallState/Installed | |
| | | wInstallDate/Date Installed | |
| Audited Files | Audited Files | name/Name | FileAudit |
| | | version/Version | |
| | | status/Status | |
| | | vendor/Vendor | |
| | | product/Product | |
| | | prodvers/Product Version | |
| | | scanfor/Scanned | |
| | | file_date/File Date | |
| | | file_size/File Size | |
| | | mtime/Modified | |
| | | file_type/File Type | |
| | | path/Path | |
| Configuration Summary for Windows | O/S Configuration | mtime/Modified | rWin32_OperatingSystem |
| | | _wOS/OS | |
| | | wRegisteredUser/Registered User | |
| | | wOrganization/Organization | |
| | | wSerialNumber/S/N | |
| | | wSystemDirectory/Sys Dir | |
| | | WtotalPageFileSpace/PageFileSize (mb) | |
| | Hardware | manufacturer/Manufacturer | rWin32_ComputerSystemProduct |
| | | _model/Model | RWin32_systemEnclosure |
| | | _wSNTag/S/N | RWin32_Processor |
| | | wManufacturer,wCurrentClockSpeed/Processor | RWin32_LogicalMemoryConf |
| | | wTotalPhysicalMemory/Physical Memory (MB) | rWin32_Computer System |
| | | wSystemType/System | rWin32_Bios |
| | | _wBios / Bios | |
| | | _wKybd/keyboard | RWind32_Keyboard |
| | | _wMouse/Mouse | rWin32_PointingDevice |
| | | _wVideo/Video/Video | rWin32_VideoController |
| | | _wDriverName/Printer | rWin32_Printer |

| Action | Displayed Table Title | Columns Queried /Display Name | Tables Queried |
|---------------|-----------------------------|-----------------------------------|-------------------------|
| | | _WSerialPort/Serial Ports | rWin32_SerialPort |
| | | _wParallelPort/Parallel Ports | rWin32_ParallelPort |
| | Network Adapter information | Wbem-show-network.tsp | |
| | Disk Drive Information | Wbem-show-drives.tsp | |
| | Environment | Wbem-show-environment.tsp | |
| | Windows Services | Wbem-show-services.tsp | |
| | Device Configuration | Ctime/Created | Device Config |
| | | Mtime/modified | |
| | | Os/OS | |
| | | Os_level/ OS Level | |
| | | Sysdrv/Sys Drive | |
| | | Sysdrv_total/ Sys Drive Size (MB) | |
| | | Sysdrv_free/Sys Drive Free (MB) | |
| | Software (AGENT and | Person/Person | Device Config |
| | | Organization/Organization | |
| | | Language/Language | |
| | | Protocol/Protocol | |
| | | Timeout/Timeout | |
| | | Trace/Trace | |
| | | Edmsys/Sys Dir | |
| | | Edmlib/Lib Dir | |
| | | Edmlog/Log Dir | |
| | Hardware | Ipaddr/ IP Address | Device Config |
| | | Macaddr/ MAC Address | |
| | | Bios/ Bios | |
| | | Cpu/CPU | |
| | | Memory/Mem (MB) | |
| | | Keyboard/Keyboard | |
| | | Mouse/Mouse | |
| | | Video/Video | |
| | | N_serial/Serial(#) | |
| | | N_Parallel/Parallel (#) | |
| | | N_PRINTER/printer (#) | |
| WBEM Features | Audited Features | wProductName/Product Name | RWind32_SoftwareFeature |

| Action | Displayed Table Title | Columns Queried /Display Name | Tables Queried |
|---------------------------|-------------------------|---|------------------------|
| | | mtime/Modified | |
| | | WInstallDate/Date Installed | |
| | | wVendor/Vendor | |
| | | wVersion/Version | |
| Installed Applications | Audited Files | Show-fileaudit.tsp | |
| | Installed Applications | WFileDescription / Application Name | RNVD_Installed_Apps |
| | | Mtime / Modified | |
| | | WPath / Path | |
| | | WoriginalFileName / Executable | |
| | | WFileVersion / Executable Version | |
| | | WcompanyName ? Vendor | |
| | | WproductName / Product Name | |
| | | WProducttVersion / Version | |
| | Add/Remove Applications | Installed-uninstall-apps.tsp | |
| WBEM Elements | Audited Elements | mtime/Modified | rWin32_SoftwareElement |
| | | wName/Name | |
| | | wVersion/Version | |
| | | wInstallDate/Date Installed | |
| | | wManufacturer/Manufacturer | |
| | | wPath/Path | |
| PDA Devices | PDA Devices | mtime/Modified | rNVD_PDASystem |
| | | wName/Name | |
| | | wDescription/Type | |
| | | wStatus/Status | |
| WBEM PDA Config | Configuration | Mtime/Modified | rCIM_OperatingSystem |
| | | wCaption, wVersion / OS | |
| | | wFreePhysicalMemory/Free Physical Memory (MB) | |
| | | wTotalVirtualMemorySize/Total Virtual Memory (MB) | |
| | | wFreeVirtualMemorySize/Free Virtual Memory (MB) | |
| | PDA Installed Products | Show-pda-inst-prod.tsp | RCIM_OperatingSystem |
| wbem-show-environment.tsp | Environment | mtime / Modified | rWin32_Environment |

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

Table 25 General Reporting – Detailed Reports

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

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

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

Table 26 History Reporting – Detailed Reports

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

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

Table 27 Summary Reporting

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

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

Table 28 Inventory Reporting – Multicast Detail Reporting

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

Index

\$

\$MACHINE, 41, 43

\$USER, 41, 43

.

.NET

installing, 52

.NET Beta, 52

.NET Framework Setup wizard, 52

A

ACCESSDT attribute, 81

accessing Reporting Server, 140

ACCESSTM attribute, 81

ACTION attribute, 76, 84, 98

ADDLOCAL argument, 37, 47, 56

agent computer, definition, 18

Agent Installation Wizard, 50

AIP, 33

ALWAYS attribute, 125

Application class, 65

Application Manager, 15

Application Self-service Manager, 15

APPSVC attribute, 126

ARGS section of Install.ini, 40

args.xml, 33, 40, 42

ARPNOREMOVE argument, 37

askconfirm argument, 41

ATTRIBUTE attribute, 81

AUDIT domain, 64

Audit Multi Files, 95

Audit Multi Files package, 99

Audit Packages class, 65

AUDIT.FILE class

ACTION, 76

GROUP, 78

INTERNAL, 79

LANGUAGE, 79

LOCATION, 80

NAME, 80

ORGNAME, 79

OUTPUT, 77

PRODUCT, 79

PRODVERS, 79

SCANFOR, 76

TYPE, 78

VENDOR, 79

VERSION, 79

ZCRCINFO, 80

ZINIT, 80

ZMD5INFO, 80

ZRSCCRC, 79

ZRSCMFIL, 79

ZRSCMMEM, 79

ZRSCOBJN, 80

ZRSCPADM, 80

ZRSCSRC, 80

ZRSCSTYP, 79

ZRSCVLOC, 79

ZVERINFO, 78

AUDIT.FILE class instance, 76

AUDIT.FILE instance, 78

AUDIT.PACKAGE class, 112

AUDIT.PACKAGE instance, 65

AUDIT.WBEM class, 83

AUDIT.WBEM instance

ACTION, 84

CLASS, 85

- CNDITION, 85
- NAME, 86
- NAMESPACE, 85
- OUTPUT, 85
- PROPERTY, 85
- TYPE, 86
- AUDIT.ZSERVICE class, 115
- AUDIT.ZSERVICE instance, 65
- auditing files, 76
- Auditing Services
 - Audit Multi Files, 95
 - CE PDA XML Inventory, 95
 - default, 95
 - Delete Discovered Application Component, 95
 - Individual File Audit, 95
 - NVDM Discovery of Applications, 95
 - Palm PDA XML Inventory, 95
 - RIM Reporting, 96
 - Unix File Scan Audit, 96
 - Unix Hardware Inventory, 96
 - Unix Software Inventory Audit, 96
 - WBEM MSI Based Applications, 96
 - WBEM Running Services, 97
 - WBEM Scan For Hardware, 97
 - WBEM Scan with Condition Statement, 97
 - WBEM Stopped Services, 97
 - WBEM System Drivers, 97
 - WBEM Windows Services, 97
 - Windows System DLL, 97

B

- BEHAVIOR class, 66
- Behavior Services class, 66

C

- cacert.pem, 43
- CE PDA XML Inventory, 95
- channelname argument, 41
- CIM. *See* Common Information Model
- CLASS attribute, 85
- clean machine, definition, 18

- Client Automation agent
 - installation program, 32
 - installing, 50
 - command line, 46
 - logon script, 49
 - modifying the installation
 - command line, 56
 - Installation Wizard, 56
 - removing
 - command line, 53
 - Installation Wizard, 53
 - repairing
 - command line, 55
 - Installation Wizard, 54
 - system requirements, 32
- Client Automation agent, description, 18
- Client Automation agents
 - feature settings, 51
- Client Methods class, 66
- Client Operations Profiles, 45
- CMETHOD class, 66
- CNDITION attribute, 85
- COLLECT attribute, 67
- command line
 - installing
 - Client Automation agent, 46
 - modifying
 - Client Automation agent installation, 56
 - removing
 - Client Automation agent, 53
 - repairing
 - Client Automation agent, 55
 - silent installation, 49

- Common Information Model, 20
 - definition, 18

- connect.bak, 42
- CREATE method, 126
- CUSTOM, 41, 43

D

- DATE attribute, 82

- default_catalog argument, 41
- default_catalog_only argument, 41
- DEFERRED, 124
- Delete Discovered Application Component, 95
- DELETE method, 126
- Desktop class, 66
- DESKTOP class, 66
- detecting an internet proxy, 61
- Differenced Audit on Install and Update, 108
- Differenced Audit on Install, Verify, and Update, 108
- DIRPATH attribute, 82
- DISCOVER attribute, 45
- discovering an internet proxy, 61
- Disk Cost, 51
- Distributed Management Task Force, 21
- document changes, 4, 5
- DOMAIN attribute, 126
- DRIVE attribute, 82

E

- EXECUTE.REXX, 132

F

- file auditing, 76
- File class, 66
- FILE class, 66
- File Scanner class, 66
- File Scanner Filters class, 66
- FILEAUDT object, 80, 81, 102, 103
 - ACCESSDT, 81
 - ACCESSTM, 81
 - ATTRIBUTE, 81
 - DATE, 82
 - DIRPATH, 82
 - DRIVE, 82
 - FULLPATH, 82
 - PAHTCRC, 82

- RESOLVE, 82
- SIZE, 82
- STATUS, 82
- TIME, 82
- XTENSION, 82

- FILENEW object, 103
- FILEPREV object, 101
- FILESCAN class, 66
- FILTER class, 66
- filtering patch reports, 143
- Full Audit on Install and Update, 108
- Full Audit on Install, Verify and Update, 108
- FULLPATH attribute, 82

G

- GATEWAY attribute, 90
- GROUP attribute, 78

H

- HALCOMP attribute, 90
- HALDATE attribute, 90
- HALFNAME attribute, 91
- HALFVER attribute, 91
- HALINAME attribute, 91
- HALLANG attribute, 91
- HALPNAME attribute, 91
- HALPVER attribute, 91
- HALSIZE attribute, 91
- HPCA Core, 14, 26
- HPCA Satellite, 14, 26
- HPCAE-MgmtAppsxx.msi, 33, 36, 55
- HTTP object transfer, 42

I

- identification argument, 41, 50
- IDMLIB directory, 38, 40, 61
 - starting, 43

- IDMLOG, 41
- IDMSYS directory, 39
- IMMEDIATE, 124
- Individual File Audit, 95
- Install.ini, 33, 39, 48
 - ARGS section, 40, 41
 - OBJECTS section, 44
 - preparing, 34
 - PROPERTIES section, 36, 37, 42
- installation program
 - Client Automation agent, 32
- installation wizard
 - installing
 - Client Automation agent, 50
 - modifying
 - Client Automation agent, 56
 - removing
 - Client Automation agent, 53
 - repairing
 - Client Automation agent, 54
- INSTALLDIR argument, 37, 50
- installed .NET, 52
- installing
 - Client Automation agent
 - command line, 46
 - Installation Wizard, 50
 - logon script, 49
- INTERNAL attribute, 79
- internet proxies, 60
- internet proxy, 61
- Inventory Database Tables, 72
- Inventory database using SQL Server, creating, 27
- Inventory Manager
 - description, 14
- Inventory Options class, 66, 67
- inventory reports, 140
- Inventory Scanners, 66
- inventorydata, 28
- inventorytemp, 28

- IPADDR attribute, 126
- IPADDR## attribute, 91

L

- LADAPT## attribute, 91
- LANGUAGE attribute, 79
- local AIP, enabling, 60
- LOCATION attribute, 80
- log argument, 41
- logon script, installing the Client Automation agent, 49
- logonpanel argument, 42
- logsize argument, 42

M

- managerurl argument, 42
- Messaging Server, 18
 - overview, 15
- MGRNAME attribute, 126
- MGRVLOC class, 67
- MGRVLOC instance, 79
- Microsoft SQL server, 27
- Modify option, 56
- modifying
 - Client Automation agent installation
 - command line, 56
 - Installation Wizard, 56
- MSIDRVR, 60
- MSIEXEC, 55

N

- NAME attribute, 68, 80, 86, 125
- NAMESPACE attribute, 85
- NVD.INI file, 41
- NVDARG, 58
- NVDENABLEIP argument, 38, 42
- NVDENABLEPROXY argument, 38

- NVDENABLESHORTCUT argument, 38
- NVDENABLEUSER argument, 38
- NVDINSTALLPATCH, 46
- NVDINSTALLRAM, 37, 46
- NVDINSTALLRIM, 37, 46
- NVDINSTALLROM, 46
- NVDINSTALLRSM, 37, 46
- NVDLOCALNOTIFYONLY argument, 39
- NVDM Discovery of Applications, 95
- NVDMMAINTDIR argument, 39
- NVDNOTIFYINTERACT argument, 39
- NVDOBJ, 58
- NVDPOSTCAPATH argument, 40, 59
- NVDPRECAPATH argument, 39, 57
- NVDRADTRAYSTART argument, 39
- NVDRAMCONNECT argument, 39
- NVDRAMSHORTCUT argument, 38
- NVDRAMSTARTMENUSHORTCUT argument, 39
- NVDREDIRECTORINTERACT argument, 39
- NVDSCHEDULERINTERACT argument, 39
- NVDSHORTCUT argument, 38
- NVDSTARTMENUICON argument, 38
- NVDSTARTWMICFGMGR argument, 38

O

- object
 - overriding property settings, 58
- OBJECTS section of Install.ini, 44
- Oracle
 - roles and system privileges, 29
- Oracle Inventory database, creating, 28
- ORNAME attribute, 79
- OUTPUT attribute, 77, 85

P

- PACKAGE class, 65

- Palm PDA XML Inventory, 95
- passport registration, 7
- Path class, 66
- PATH class, 66
- PATHCRC attribute, 82
- PENDING attribute, 132
- PERIODIC, 124
- PINGCNT attribute
 - specifying, 133
- PINGDLAY attribute, 127
 - specifying, 133
- Portal
 - description, 18
- POST_WBEM method
 - disabling, 87
- post-install script, 34, 40, 58
- pre-install script, 34, 39, 57
- PRIMARY.AUDIT, 94
- PRODUCT attribute, 79
- PRODVERS attribute, 79
- PROPERTIES section of Install.ini, 36, 42
- PROPERTY attribute, 85
- providername argument, 42
- PROXADDR attribute, 45
- PROXYINF object, 38
- PROXYINF.EDM, 61

Q

- QMSG, 87
- QMSG.EXE, 87

R

- RADIA.MSI, 48
- RADSETUP_COP attribute, 45
- radskman, 132
- REBOOT attribute, 91

- REBOOTD attribute, 91
- redirect argument, 42
- redirect tag, 42
- Registry class, 66
- REGISTRY class, 66
- REMOVE argument, 47, 54, 56
- Remove button, 37
- Remove option, 53
- removing
 - Client Automation agent
 - command line, 53
 - Installation Wizard, 53
- Repair option, 54
- repairing
 - Client Automation agent
 - command line, 55
 - Installation Wizard, 54
- Reporting Server, 16, 18
 - environment, 16
 - filtering patch reports, 143
 - overview, 16
- REQUEST attribute, 126
- resolutionmanager argument, 42, 51
- resolutionport argument, 43, 51
- RESOLVE attribute, 82
- RIM Reporting, 96
- RIM_REPORTING service, 136
- RIMDIFF method, 76, 102, 103
- RIMFSCAN method, 76, 102, 103
- RIMOPTS, 107
 - customizing, 134
 - Default, 107
 - Differenced Audit on Install and Update, 108
 - Differenced Audit on Install, Verify, and Update, 108
 - Full Audit on Install and Update, 108
 - Full Audit on Install, Verify and Update, 108
- RIMOPTS class, 66, 67

- RIMOPTS instance, 68
- RIMSFSCAN, 22
- RIMWBEM, 22
- RIMWBEM method, 83
- root_catalog_name argument, 43
- RUNEXEC attribute, 68, 136
- RUNSYNC attribute, 126

S

- SCANFOR attribute, 76, 97
- SCANNER class, 66
- Scheduling (TIMER) class, 122, 124
 - _ALWAYS_, 125
 - APPSVC, 126
 - DOMAIN, 126
 - IPADDR, 126
 - MGRNAME, 126
 - NAME, 125
 - PINGDLAY, 127
 - REQUEST, 126
 - RUNSYNC, 126
 - SOCKET, 126
 - ZCREATE, 126
 - ZDELETE, 126, 127
 - ZNOPING, 126
 - ZOBJPRI, 124
 - ZRSCCMDL, 125
 - ZSCHDEF, 124
 - ZSCHFREQ, 125
 - ZSCHMODE, 124
 - ZSCHTYPE, 124
 - ZSTOP, 124
 - ZSVCOID, 125
 - ZUPDATE, 126
 - ZVERIFY, 126
- Scheduling class, 67
- script
 - post-install, 58
 - pre-install, 57
- Set User window, 41
- setup.exe, 33

- location, 48
- silent installation, 47
 - command line, 49
- SIZE attribute, 82
- SOCKET attribute, 126
- sslmanager argument, 43
- sslport argument, 43
- startdir argument, 43
- STATUS attribute, 82
- SUBNET## attribute, 91
- system requirements
 - Client Automation agent installation, 32
- SYSTEM.PROCESS.WBEMAUDT
 - disabling, 87
- SYSTEM.ZMETHOD.POST_WBEM
 - disabling, 87

T

- tablespace, creating, 28
- TIME attribute, 82
- TIMER class, 67, 122
- TYPE attribute, 78, 86

U

- UIOPTION attribute, 44
- Unix File Scan Audit, 96
- Unix Hardware Inventory, 96
- Unix Software Inventory Audit, 96
- UPDATE method, 126
- USEPROXY attribute, 45
- using an internet proxy, 61

V

- VENDOR attribute, 79
- VERIFY method, 126
- VERSION attribute, 79

- Virtual Mgr Location class, 67

W

- WBEM auditing, 83
- WBEM class, 67
- WBEM MSI Based Applications, 96
- WBEM namespace, 22, 83
- WBEM object variables
 - ZOBJCID, 86
 - ZOBJCLAS, 86
 - ZOBJCIRC, 86
 - ZOBJDATE, 86
 - ZOBJDOMN, 86
 - ZOBJID, 86
 - ZOBJNAME, 86
 - ZOBJPCLS, 86
 - ZOBJPID, 86
 - ZOBJRCRC, 86
 - ZOBJRSIZ, 86
 - ZOBJTIME, 87
 - ZRSCSRC, 87
- WBEM Running Services, 97
- WBEM Scan for Hardware, 97
- WBEM Scan with Condition Statement, 97
- WBEM Stopped Services, 97
- WBEM System Drivers, 97
- WBEM Windows Services, 97
- WBEMAUDT object
 - processing, 87
- WBEMAUDT process
 - disabling, 87
- Web-Based Enterprise Management, 21
- WEBM. *See* Web-Based Enterprise Management
- Windows Installer log, 47
- Windows Management Instrumentation, 21
 - description, 19
- Windows Registry
 - scanning, 66
- Windows System DLL, 97

Windows Vista Readiness Reports, 143
WMI. *See* Windows Management Instrumentation
 used for registry scan, 66
WMI repository, 21

X

XML tags, 40
xmlCIM, 21
XTENSION attribute, 82

Z

ZCONFIG object, 88
ZCONFIG variable, 88
ZCRCINFO, 80
ZCRCINFO attribute, 80
ZCREATE attribute, 126
ZDELETE attribute, 126, 127
ZDSTSOCK attribute, 44, 58
ZGATEWAY attribute, 91
ZHDWBIOS attribute, 91
ZHDWCDDR attribute, 91
ZHDWCOMP attribute, 91
ZHDWCPU attribute, 91
ZHDWFPU attribute, 91
ZHDWIPAD attribute, 91
ZHDWKYBD attribute, 91
ZHDWLANA attribute, 91
ZHDWLANG attribute, 91
ZHDWMEM attribute, 91
ZHDWMEMF attribute, 91
ZHDWMOUS attribute, 91
ZHDWNET# attribute, 91
ZHDWNNET attribute, 91
ZHDWOS attribute, 92
ZHDWOSCL attribute, 92

ZHDWOSDB attribute, 92
ZHDWOSOG attribute, 92
ZHDWOSOW attribute, 92
ZHDWOSSR attribute, 92
ZHDWPA## attribute, 92
ZHDWPPAR attribute, 92
ZHDWPPRN attribute, 92
ZHDWPSEER attribute, 92
ZHDWVIDEO attribute, 92
ZHDWVMSI attribute, 92
ZHDWVRES attribute, 92
ZHDWXPAG attribute, 92
ZHWCPU01 attribute, 92
ZHWFPU01 attribute, 92
ZINIT attribute, 80
ZIPADDR attribute, 44
ZMASTER object, 49
ZMASTER.EDM, 49
ZMD5INFO attribute, 80
ZMODEM attribute, 92
ZNOPING attribute, 126
 specifying, 132
ZNTFPORT attribute, 44
ZNTFYSEC attribute, 44
ZOBJCID attribute, 86
ZOBJCLAS attribute, 86
ZOBJCRC attribute, 86
ZOBJDATE attribute, 86, 92
ZOBJDOMN attribute, 86
ZOBJID attribute, 86
ZOBJNAME attribute, 86, 92
ZOBJPCLS attribute, 86
ZOBJPID attribute, 86
ZOBJPRI attribute, 124

ZOBJPRI attribute, 124
ZOBJRCRC attribute, 86
ZOBJRSIZ attribute, 86
ZOBJTIME attribute, 87, 92
ZRSCCMDL attribute, 125, 131, 132
 specifying, 131
ZRSCCRC attribute, 79
ZRSCMFIL attribute, 79, 117
ZRSCMMEM attribute, 79
ZRSCOBJN attribute, 80
ZRSCPADM attribute, 80
ZRSCSRC attribute, 80, 87
ZRSCSTYP attribute, 79
ZRSCVLOC attribute, 76, 79
ZSCHDEF attribute, 124, 129, 131
ZSCHFREQ attribute, 125
 specifying, 131
ZSCHMODE attribute, 124
ZSCHTYPE attribute, 124
 specifying, 131

ZSERVICE class, 65
ZSTOP attribute, 124
ZSUBNET attribute, 92
ZSVCMO attribute, 117, 138
ZSVCOID attribute, 125
ZSVCTYPE attribute, 68, 107
ZTASKEND, 87
ZTASKEND rexx, 87
ZTIMEO attribute, 44
ZTIMEQ object, 124, 131, 132
ZTRACE attribute, 45
ZTRACEL attribute, 45
ZUPDATE attribute, 126
ZUSERID attribute, 45, 92
ZVERIFY attribute, 126
ZVERINFO attribute, 78
ZVRFYUID attribute, 45

