II Server Inventory

The Server Inventory component of Novell® ZENworks® 6.5 Server Management enables you to collect hardware and software inventory information from local and remote servers of your enterprise. This inventory information is scanned and stored in a database that can be accessed by the ZENworks administrator.

From Novell ConsoleOne®, you can view the complete hardware and software inventory of the servers. You can also query the centralized database of the servers.

The following sections provide information on the features and tasks of Server Inventory:

12 Understanding Server Inventory

The Server Inventory component of Novell® ZENworks® 6.5 Server Management gathers hardware and software inventory information from Novell NetWare® and Windows* servers in your enterprise and stores into a centralized database. Using this database, the network administrator can view and query for complete inventory information for the enterprise.

The inventory information can be useful to help you make business decisions on how to manage servers. The following are some of the business decisions that you can make once you have obtained the inventory information:

- Servers that need new applications
- Servers that need updated hardware and drivers
- Servers that conform to the corporate hardware and software standards

This chapter provides a basic overview of Server Inventory. It contains the following information:

Server Inventory Terminology

The following brief glossary provides basic definitions of Server Inventory terms:

Inventoried server: A server whose hardware and software information you want to scan and maintain in a central repository. To gather complete hardware and software inventory for a server, you must install the Inventory Agent on that server.

Inventory server: A server where you run the Inventory service. This server can run any other ZENworks 6.5 Server Management services also. The Inventory server collects the inventory information from a group of associated inventoried servers and stores it into the Inventory database. If you want to collect the inventory for the Inventory server, you must install the Inventory Agent on that Inventory server.

Inventory database: A repository of inventory information of all the inventoried servers.

Database server: A server running Sybase*, Oracle*, or MS SQL where your Inventory database is mounted. The database can run on an Inventory server or on a different server.

Management console: A Windows workstation or server running Novell ConsoleOne® with Server Inventory ConsoleOne snap-ins installed. The management console provides the interface to administer the inventory system.

eDirectory Tree: The Novell eDirectory™ tree consists of eDirectory objects such as multiple levels of organizational units, users, groups, and other network resources. This hierarchical structure is referred to as the eDirectory tree in this document. For more information, see the Novell eDirectory documentation Web site (http://www.novell.com/documentation).

Inventory tree: A logical tree depicting the transmission of the inventory information from the inventoried servers and the Inventory servers to the centralized enterprise Inventory database.

Standalone Server: An Inventory server that has an Inventory database and inventoried servers attached to it.

Leaf Server: The lowest-level Inventory server in the inventory tree hierarchy. This server has one or more inventoried servers attached to it and can have an Inventory database attached to it. This Inventory server collects the inventory information from the inventoried servers attached to it and moves the information to the next-level Inventory server.

Intermediate Server: The Inventory server for moving the inventory information from the lower-level Inventory servers up the Inventory server hierarchy. This server can have either inventoried servers or an Inventory database, or both attached to it.

Root Server: The highest-level Inventory server in the inventory tree hierarchy. This server has the Inventory database that contains the inventory information of all the lower-level Inventory servers. At the Root Server level, you can view complete inventory information for the entire enterprise. This server can have inventoried servers attached to it.

Site: A site is typically a geographical location. There can be multiple sites your enterprise.

Software Dictionary or Dictionary: The software dictionary contains a list of software identifiers and rules. Each software identifier identifies a particular product installed on an inventoried server.

Software Identifiers: An entry that identifies a software product is called as a software identifier. Each software identifier has a set of file matching attributes and corresponding software information attributes. During the Inventory scan, the scanner reads the attributes from the file headers, and if these attributes match the attributes configured in the dictionary, the information in the corresponding software information attributes is stored in the Inventory database.

Software Dictionary Rule: A software dictionary rule represents a set of conditions that control the scope of scanning process.

Unidentified Software: The software dictionary might not identify all the software installed in your network. The software that are not listed in the dictionary are called Unidentified software.

Overview of Server Inventory Components

Before setting up the ZENworks 6.5 Server Inventory, you should understand the inventory components that interact together to perform inventory functions.

Server Inventory uses the following components:

- “Inventory Scanners” on page 419
- “Inventory Components on Inventory Servers” on page 419
- “Inventory Database” on page 420
- “Management Console” on page 420

Inventory Scanners

Platform-dependent scanners determine the hardware and software configurations of the inventoried servers. These scanners are located at the inventoried servers. When executed on the inventoried servers, the scanners collect the inventory information. The inventory information is subsequently transferred to the Inventory server and processed.

Using the Server Inventory policy, you can configure the scan settings for scheduling the scan on the inventoried servers and customize hardware scanning. From the Inventory Service object (Inventory Service_*server_name*), you can specify the location of the inventory information, and also customize software scanning using the Software Dictionary snap-ins.

For more information about the Inventory scanner, see “Understanding the Inventory Scanner” on page 496.

Inventory Components on Inventory Servers

The Inventory server components process the inventory information. The following components are Java* programs that work identically on NetWare and Windows Inventory servers:

- Scan Collector

  The Scan Collector collects the inventory information from the Inventory agent and stores them in an appropriate directory at the Inventory server. The inventory information is transferred using the XML-RPC protocol.

- Selector

  The Selector processes the inventory information and places the information in appropriate directories. For more information, see “Understanding the Selector” on page 508.

- Sender and Receiver

  The Sender on the Inventory server compresses the inventory information and then transfers it from the lower-level Inventory server to the Receiver on the higher-level Inventory servers. By using the Roll-Up policy, you can configure the next level destination Inventory server for roll-up, and also schedule the roll-up time. For more information, see “Understanding the Sender - Receiver” on page 503.

- Storer

  The Storer stores the collected inventory information into the Inventory database. By using the Database Location policy, you can configure the properties of the Inventory Database object (Inventory database_*server_name*) and associate the database object to an Inventory server. For more information, see“Understanding the Storer” on page 509.

- Dictionary Provider and Dictionary Consumer

  All Inventory servers run the Dictionary Provider and Dictionary Consumer services.The Dictionary Consumer downloads the dictionary updates from the Dictionary Provider. For more information, see “Understanding the Dictionary Provider - Dictionary Consumer” on page 509.

Inventory Database

The Inventory database is a repository of inventory information of the inventoried servers. In Server Management, the database is a Common Information Model-based database and is implemented in Relational Database Management System (RDBMS). The database is maintained in Sybase, Oracle, or MS SQL. For more information, see “Setting Up the Inventory Database” on page 465.

Management Console

The management console is the Novell ConsoleOne, the single management tool for administration. This is a Java-based console that includes snap-ins for Server Inventory management operations.

Understanding Inventory Scanning Cycle

The Inventory scanning cycle is as follows:

1. The Inventory scanner checks whether an updated dictionary is available at its Inventory server and downloads the updated dictionary.
2. The Inventory scanner sends hardware and software information from the inventoried servers to the Inventory server as per the scan schedule.
3. The Inventory server stores the inventory information in the Inventory database.
4. At the management console, you can view and retrieve the inventory information from the Inventory database using Inventory tools such as Reporting, Summary, etc.

Understanding the Inventory Server Roles

This section describes the following roles that you can assign for an Inventory server:

- “Root Server” on page 421
- “Root Server with Inventoried Servers” on page 422
- “Intermediate Server” on page 423
- “Intermediate Server with Database” on page 424
- “Intermediate Server with Database and Inventoried Servers” on page 426
- “Leaf Server” on page 427
- “Leaf Server with Database” on page 428
- “Standalone Server” on page 429

For a quick reference table of the Inventory Server roles, see “Quick Reference Table of the Inventory Server Roles” on page 429.

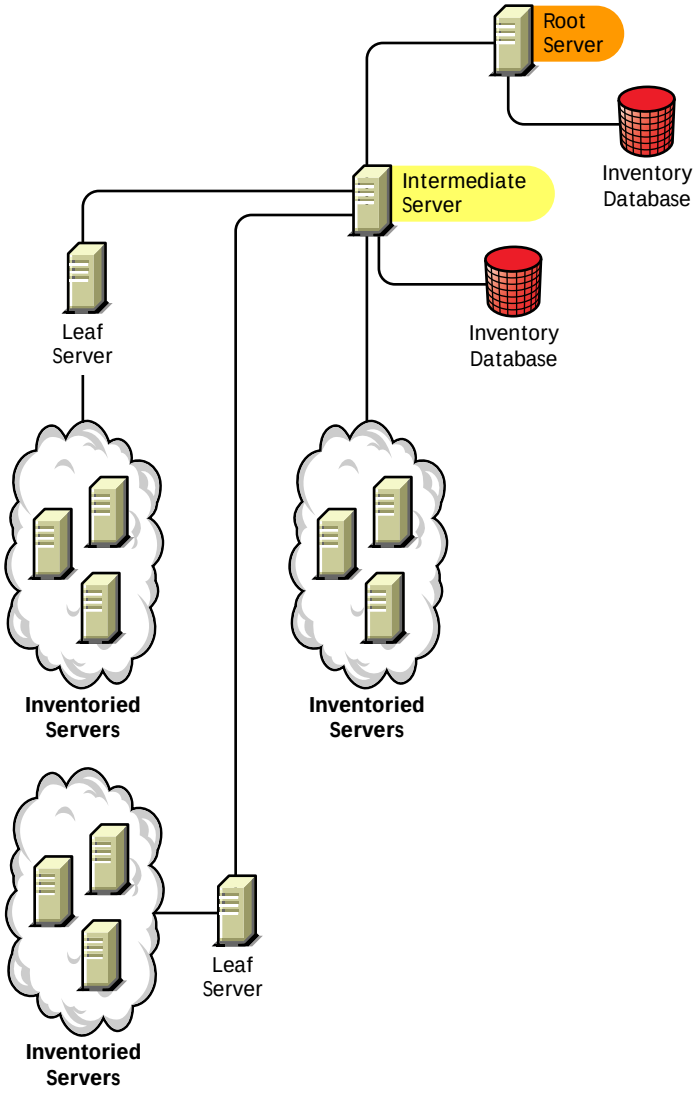
Root Server

The Root Server has the following characteristics:

- This server is the topmost Inventory server in the inventory tree hierarchy.
- This server has an Inventory database attached to it.

Choose Root Server to store the inventory information for your enterprise in a centralized database. The Inventory database at the Root Server contains the inventory information for all the lower-level Inventory servers.

The following illustration depicts Leaf Servers connected to the Intermediate Server with Database. The Intermediate Server is attached to the Root Server.



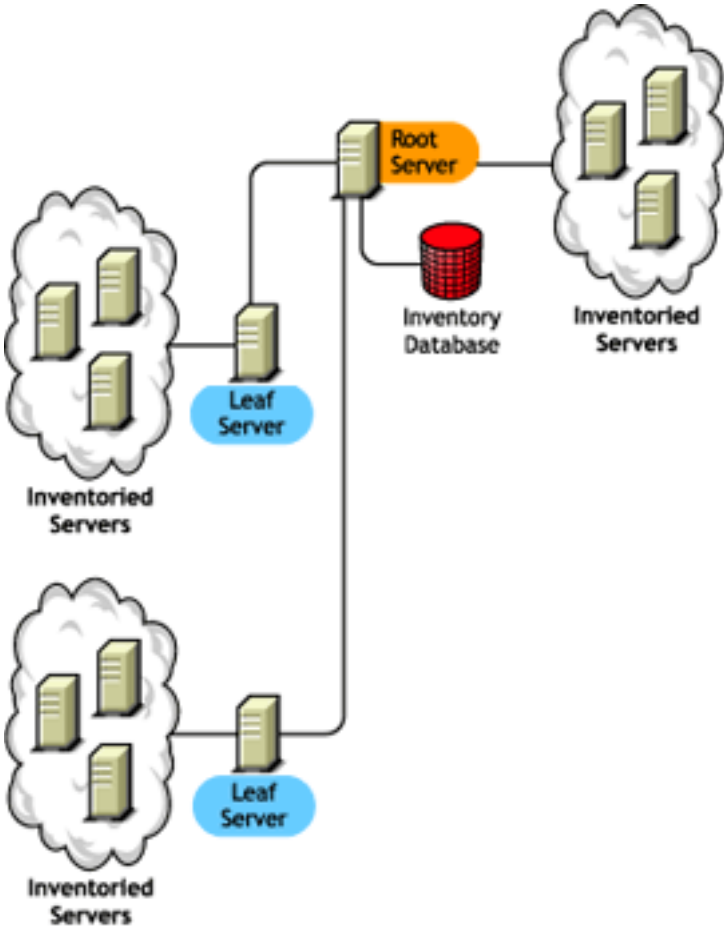
Root Server with Inventoried Servers

The Root Server with Inventoried Servers has the following characteristics:

- This server is the topmost Inventory server in the inventory tree hierarchy.
- This server has an Inventory database and inventoried servers attached to it. We recommend that you have to deploy these inventoried servers in a LAN.

Choose Root Server with Inventoried Servers if you want to store the inventory information of your enterprise in a centralized database and if you have inventoried servers in the same site as the Root Server. You can directly send the inventory information from these inventoried servers to the Root Server. The Inventory database at the Root Server with Inventoried Servers contains the inventory information for all these inventoried servers as well as for all the lower-level Inventory servers.

The following illustration depicts a Root Server with inventoried servers and Inventory database attached to it. The Leaf Servers are connected to the Root Server.



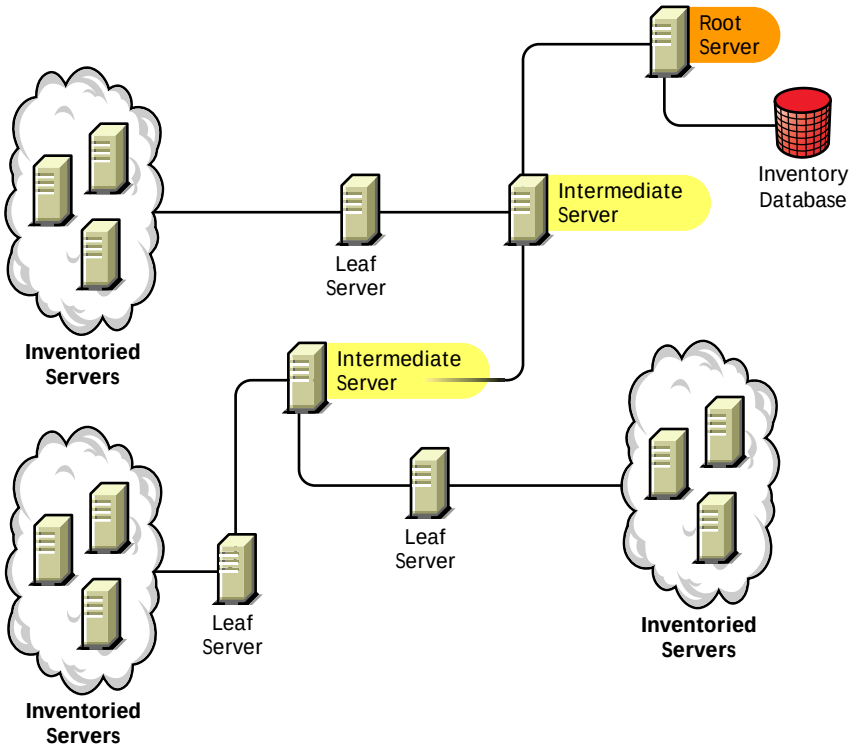
Intermediate Server

The Intermediate Server has the following characteristics:

- This Inventory server acts as a staging server for the lower-level Leaf Servers.
- This server moves the inventory information to the next-level Inventory server.
- This server does not have inventoried servers or an Inventory database attached to it.
- There can be one or more Intermediate Servers in your enterprise.

Place Intermediate Servers on sites where the link parameters change substantially. The Intermediate Server stores the scan files to the disk to make up for the difference in bandwidth and reliability.

The following illustration depicts an Intermediate Server connected to Root Server. Two Leaf Servers roll up the inventory information to the Intermediate Server. This Intermediate Server rolls up the inventory information to another Intermediate Server that is connected to the Root Server.



In this illustration, there are many Leaf Servers and Intermediate servers at different levels. The Intermediate server is a staging server for uploading the scan information to the next-level server. The last Intermediate Server is attached to the topmost Root Server. This scenario is typical if there are many Leaf Servers in different geographical locations. All the Leaf Servers move the inventory information to the Intermediate Server.

In some scenarios, the Leaf Server connects to the Intermediate Server over a WAN.

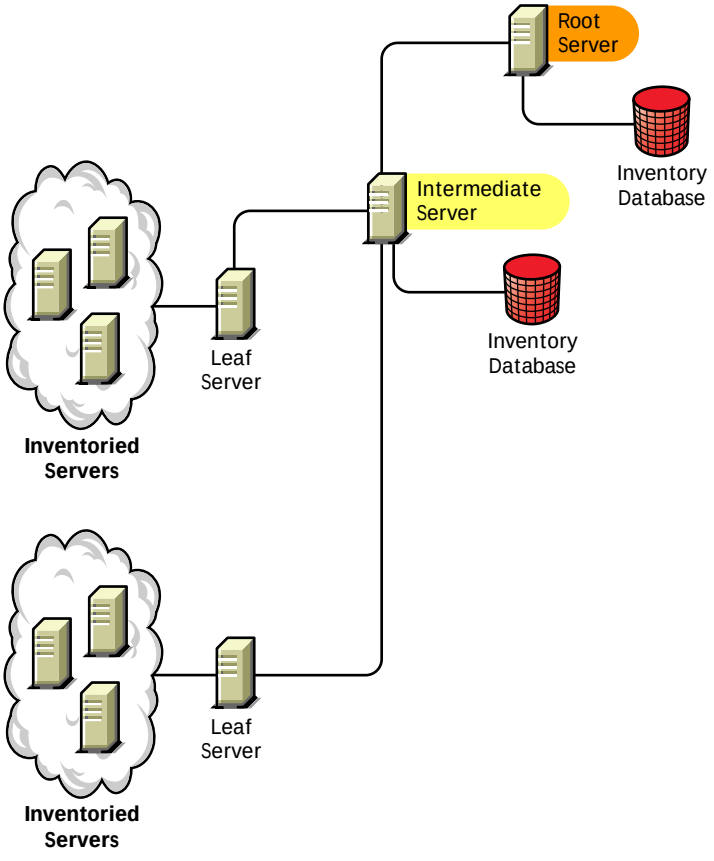
Intermediate Server with Database

The Intermediate Server with Database has the following characteristics:

- This Inventory server acts as a staging server for the lower-level Leaf Servers.
- This server moves the inventory information to the next-level Inventory server.
- This server has an Inventory database attached to it.
- There can be one or more Intermediate Servers with Database in your enterprise.

Choose Intermediate Server with Database if you want to administer an intermediate site by generating Inventory reports. The inventory information that is rolled up to this Inventory server is stored in the local Inventory database and also rolled up to the next-level Inventory server.

The following illustration depicts two Leaf Servers attached to the Intermediate Server. A consolidated inventory information of all Leaf Servers is available at the Intermediate Server level.



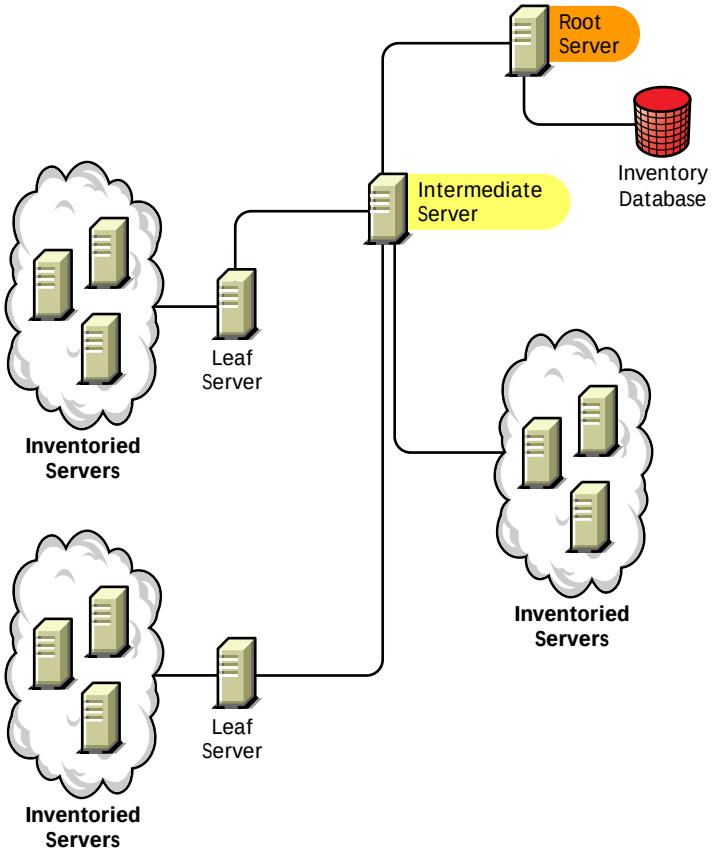
Intermediate Server with Inventoried Servers

The Intermediate Server with Inventoried Servers has the following characteristics:

- This Inventory server acts as an intermediate server for the lower-level Leaf Servers.
- This server moves the inventory information to the next-level Inventory server.
- This server has inventoried servers attached to it.
- This server does not have an Inventory database attached to it.
- There can be one or more Intermediate Servers with Inventoried Servers in your enterprise.

Choose Intermediate Server with Inventoried Servers if you want an Intermediate Server and the site having the Intermediate Server has inventoried servers, whose inventory information you want to store it at the Root Server.

The following illustration depicts two Leaf Servers attached to the Intermediate Server. This Intermediate Server also has inventoried servers attached to it.



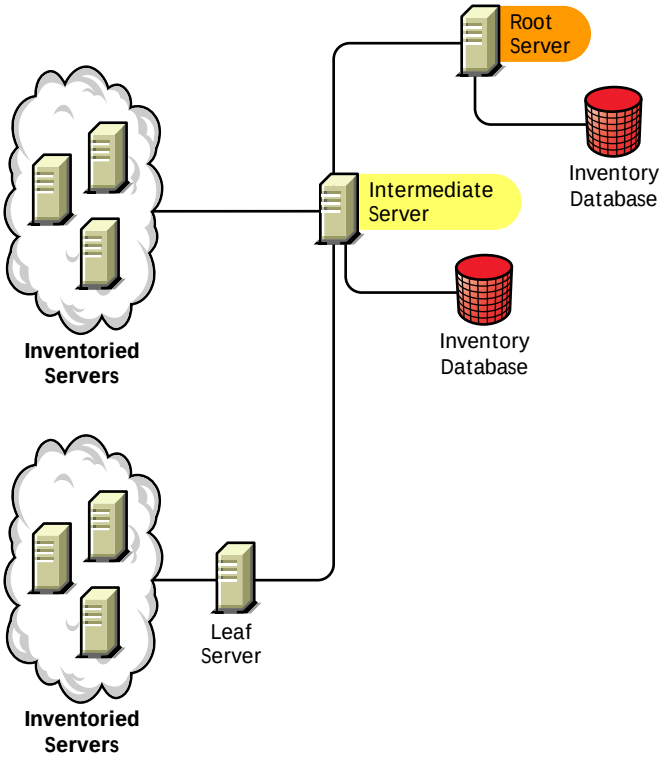
Intermediate Server with Database and Inventoried Servers

The Intermediate Server with Database and Inventoried Servers has the following characteristics:

- This Inventory server acts as an staging server for the lower-level Leaf Servers.
- This server moves the inventory information to the next-level Inventory server.
- This server has inventoried servers attached to it.
- This server has an Inventory database attached to it.
- There can be one or more Intermediate Servers with Database and Inventoried Servers in your enterprise.

Choose Intermediate Server with Database and Inventoried Servers if you want the functionalities of Intermediate Server with Database and Intermediate Server with Inventoried Servers available on the site.

The following illustration depicts two Leaf Servers attached to the Intermediate Server. The Intermediate Server has inventoried servers attached to it. A consolidated Inventory database of all Leaf Servers and the inventoried servers that are directly connected to the Intermediate Server is available at the Intermediate Server level.



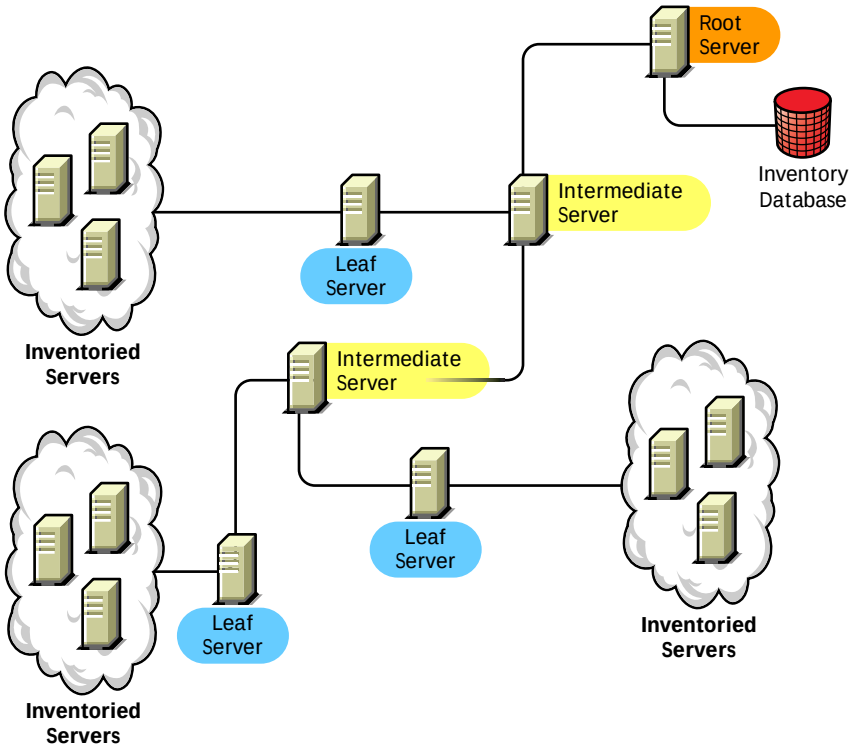
Leaf Server

The Leaf Server has the following characteristics:

- This Inventory server is at the lowest level in the inventory tree hierarchy.
- This server has inventoried servers attached to it.
- This server moves the inventory information to the next-level Inventory server.
- A simple Leaf Server does not have an Inventory database. An Inventory database is not required because there might be only few inventoried servers attached to the Leaf server.

Choose Leaf Server if you have inventoried servers at remote sites, and you want to obtain and store the inventory information from these inventoried servers in a centralized database.

The following illustration depicts many Leaf Servers attached to the Intermediate Server. The Intermediate Server is connected to Root Server. A consolidated Inventory database of all Leaf Servers is available at the Root Server level.



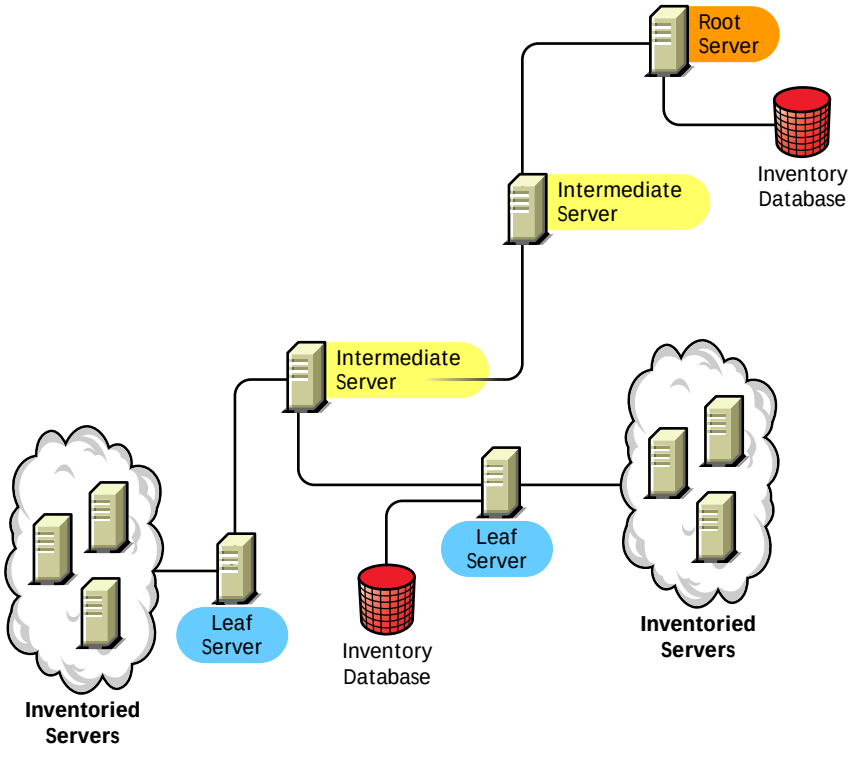
Leaf Server with Database

The Leaf Server with Database has the following characteristics:

- This Inventory server is at the lowest level in the inventory tree hierarchy
- This server has inventoried servers attached to it.
- This server moves the inventory information to the next-level Inventory server.
- This server has an Inventory database attached to it. You can assign a server as a Leaf Server with Database to maintain the inventory information for inventoried servers specific to the site.

Choose Leaf Server with Database if you want the functionalities of a Leaf Server as well as administer the site by generating Inventory reports

The following illustration depicts two Leaf Servers attached to the Intermediate Server. One Leaf Server has an Inventory database attached to it. This database contains a consolidated inventory of all inventoried servers attached to this Leaf Server.



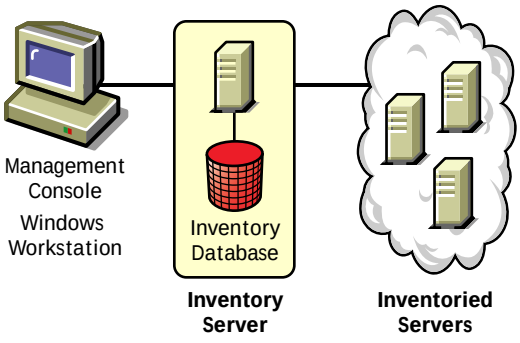
Standalone Server

The Standalone Server has the following characteristics:

- This Inventory server has inventoried servers attached to it.
- This server has an Inventory database attached to it.
- There is no roll-up of scan information and there are no requirements for Intermediate Servers and a Root Server.

Use a Standalone Server if your network is made up of a single site and you want to administer that site.

The following illustration depicts Standalone Server.



Quick Reference Table of the Inventory Server Roles

Inventory Server	Is Inventory Database Attached to the Inventory Server?	Are Inventoried Servers Attached to the Inventory Server?
Root Server	Yes	No
Root Server with Inventoried Servers	Yes	Yes
Intermediate Server	No	No
Intermediate Server with Database	Yes	No
Intermediate Server with Inventoried Servers	No	Yes
Intermediate Server with Database and Inventoried Servers	Yes	Yes
Leaf Server	No	Yes
Leaf Server with Database	Yes	Yes
Standalone Server	Yes	Yes

13 Setting Up Server Inventory

Before you install Novell® ZENworks® 6.5 Server Inventory in your working environment, you must plan and decide the hierarchy of the Inventory server tree for your enterprise. You should organize your inventory deployment based on your network constraints and information requirements.

The following sections contain detailed information to help you deploy Server Inventory in your enterprise:

Deploying Server Inventory

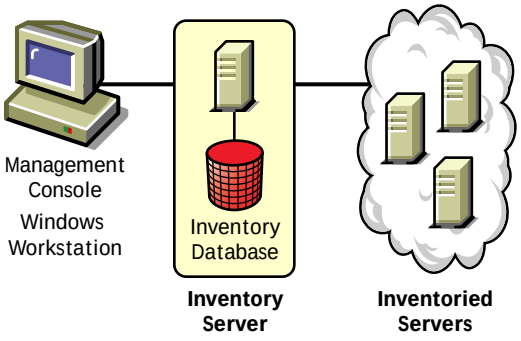
The following sections will help you to deploy Server Inventory:

IMPORTANT: The recommendations discussed in the scenarios are generic. Based on the topology of your enterprise, further refinements might become necessary.

Simple Deployment

In the example scenario, the network consists of a single site and up to 5000 inventoried servers. The Inventory server components and the database are located on a Standalone Server, and the inventoried servers send scans to the Standalone server.

This scenario is illustrated in the following figure.



The following sections contain detailed information to help you deploy Server Inventory in a single site:

1. “Recommendations for Deployment” on page 432
2. “Installing Server Inventory” on page 433
3. “Understanding the Effects of Server Inventory Installation” on page 433
4. “Configuring the Required Policies” on page 433
5. “Starting the Inventory Service” on page 433
6. “Updating the Software Dictionary” on page 433
7. “Understanding the Inventory Scanning Cycle in the Standalone Scenario” on page 434

Recommendations for Deployment

- The minimum base Inventory server configuration includes 512 MB RAM and a database cache of 128 MB.
- The transmission of inventory information to the Inventory server and storage of the inventory information into the Inventory database is an ongoing backend process that can take several hours or even more than a day.
- If many inventoried servers are attached to the same Inventory server, we recommend that you do not schedule the scan of all inventoried servers at the same time, because this will stress the Novell eDirectory™ and the Inventory services.
- Ensure that the eDirectory time synchronization radius is set within 2 seconds.
- The optimal database cache size requirement for the server could vary because of the server environment. Determine the database cache size that needs to be set by trying a range of cache sizes in the runtime environment. The default Sybase database cache size is 128 MB. For more information about improving the database performance, see Appendix H, “Performance Tips,” on page 683.

Installing Server Inventory

During the Server Inventory installation, configure the Inventory Standalone Configuration settings. For detailed information on installing Server Inventory, see "Installation on NetWare and Windows" in "Policy-Enabled Server Management Installation" in the *Novell ZENworks 6.5 Server Management Installation Guide*.

Understanding the Effects of Server Inventory Installation

For detail information on the effects of Server Inventory installation, see "Understanding the Effects of Server Inventory Installation" on page 452.

Configuring the Required Policies

Configure the Server Inventory Policy.

Starting the Inventory Service

After installing ZENworks 6.5 Server Management, the Inventory service is automatically started.

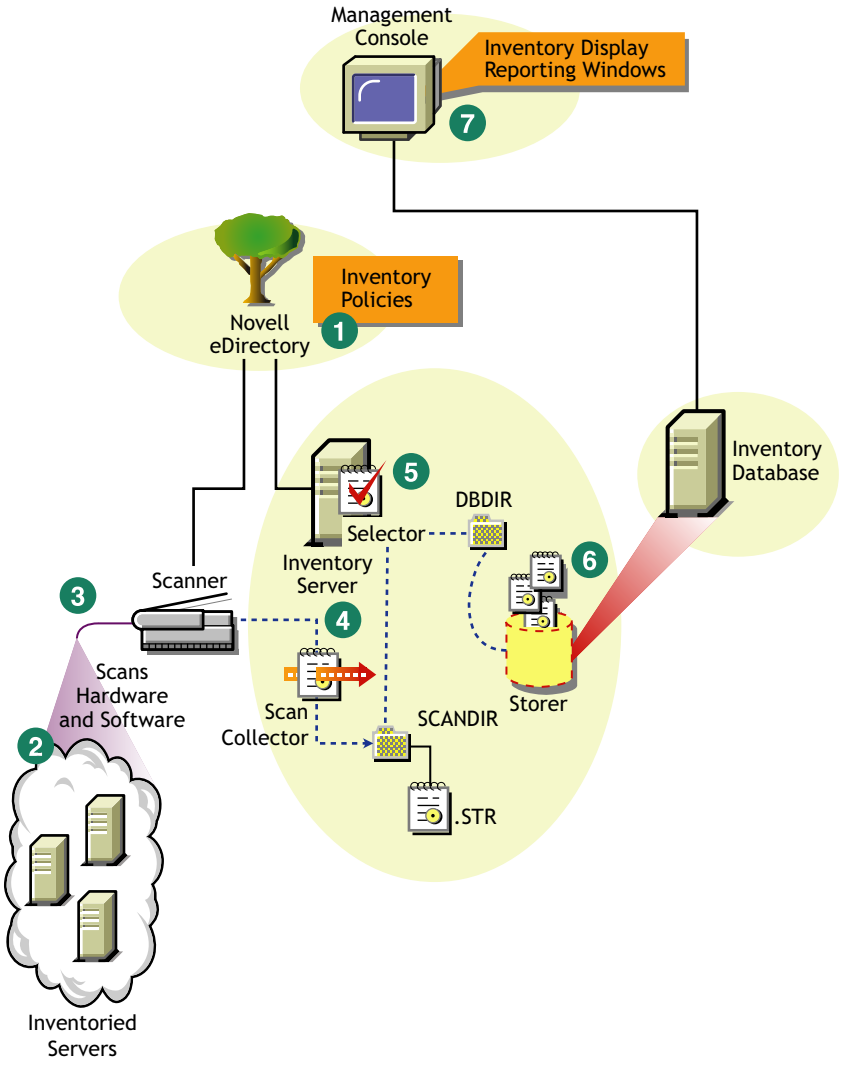
Updating the Software Dictionary

You must manually download the latest version of the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) and update the software dictionary.

NOTE: The dictionary is updated and published once every three months in this TID.

Understanding the Inventory Scanning Cycle in the Standalone Scenario

The following illustration depicts the scanning components and the inventory scanning cycle in the standalone scenario, which is explained below:



The inventory scanning cycle is as follows:

1. The inventory policies in the eDirectory define the inventory settings, such as the Inventory Service object name of the Inventory server to which the inventory information will be sent, scanning time, etc. These settings are customizable.
2. The scanner uses Policy and Distribution Services to read the inventory policies and collects the inventory information based on the policy settings.The Inventory scanner also checks whether an updated dictionary is available at its Inventory server and downloads the updated dictionary
3. The scanner stores the inventory information locally on the inventoried server. This information is transferred to the Inventory server using the XML-RPC protocol.
4. The Scan Collector receives the inventory information using the XML-RPC protocol and stores the information in the scan directory at the Inventory server. The Scan Collector uses the ZENworks Web Server to process the XML-RPC requests.

5. The Selector validates the inventory information and places the information in the Inventory database.
6. The Storer updates the database with the inventory information.
7. The ZENworks administrator views the inventory information.

Advanced Deployment

Deploying Inventory in a Single Site with More than 5,000 Inventoried Servers

In this example scenario, the network consists of a single site with more than 5000 inventoried servers. The inventory configuration consists of two or more Standalone Servers; each server receiving scans from up to 5,000 inventoried servers. All the Standalone Servers store the inventory data to a single database.

The following sections contain detailed information to help you deploy Server Inventory in a single site:

Recommendations for Deployment

- The minimum base Inventory server configuration includes 512 MB RAM and a database cache of 128 MB.
- All inventoried servers should send the inventory information to the nearest Inventory server on the LAN; policies must be created based on this information.
- The transmission of inventory information to the Inventory server and storage of the inventory information into the Inventory database is an ongoing backend process that can take several hours or even more than a day.
- If many inventoried servers are attached to the same Inventory server, we recommend that you do not schedule the scan of all inventoried servers at the same time, because this will stress the Novell eDirectory™ and the Inventory services.
- Ensure that the eDirectory time synchronization radius is set within 2 seconds.
- The optimal database cache size requirement for the server could vary because of the server environment. Determine the database cache size that needs to be set by trying a range of cache sizes in the runtime environment. The default Sybase database cache size is 128 MB. For more information about improving the database performance, see Appendix H, "Performance Tips," on page 683.

Installing Server Inventory

For detail information on installing Server Inventory, see "Installation on NetWare and Windows" in "Policy-Enabled Server Management Installation" in the *Novell ZENworks 6.5 Server Management Installation Guide*.

Understanding the Effects of Server Inventory Installation

For detail information on the effects of Server Inventory installation, see "Understanding the Effects of Server Inventory Installation" on page 452.

Configuring the Required Policy

Configure the Server Inventory Policy.

Starting the Inventory Service

After installing ZENworks 6.5 Server Management, the Inventory service is automatically started.

Updating the Software Dictionary

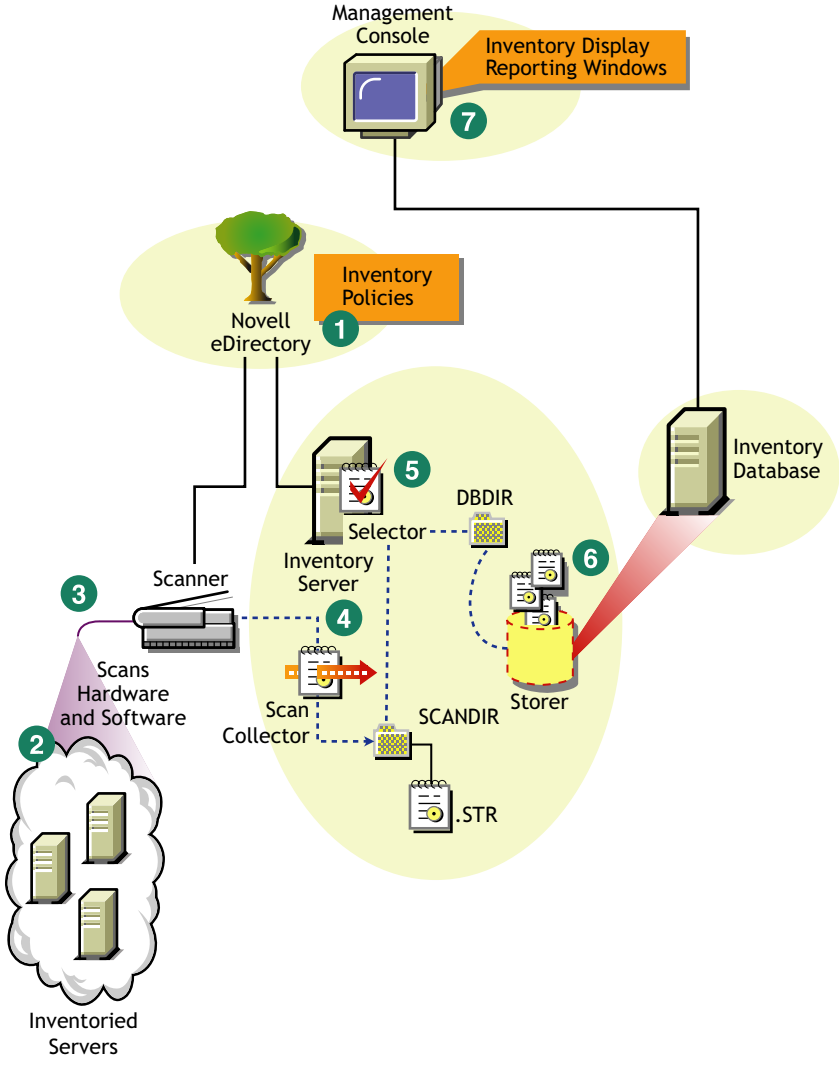
You can update the Software Dictionary in any one of the following ways:

- On each Inventory server, manually download the latest version of the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) and update the software dictionary.
- Manually download the latest version of the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) on one of the Standalone Servers and automatically distribute the dictionary from this server to all other Standalone Servers in your setup by configuring the "Configuring the Dictionary Update Policy" on page 489. For more information, see "Setting Up Distribution of Dictionary" on page 490.

NOTE: The dictionary is updated and published once every three months in this TID.

Understanding the Inventory Scanning Cycle in the Standalone Scenario

The following illustration depicts the scanning components and the inventory scanning cycle in the standalone scenario, which is explained below:



The inventory scanning cycle is as follows:

1. The inventory policies in the eDirectory define the inventory settings, such as the Inventory Service object name of the Inventory server to which the inventory information will be sent, scanning time, etc. These settings are customizable.
2. The scanner uses Policy and Distribution Services to read the inventory policies and collects the inventory information based on the policy settings.The Inventory scanner also checks whether an updated dictionary is available at its Inventory server and downloads the updated dictionary
3. The scanner stores the inventory information locally on the inventoried server. This information is transferred to the Inventory server using the XML-RPC protocol.
4. The Scan Collector receives the inventory information using the XML-RPC protocol and stores the information in the scan directory at the Inventory server. The Scan Collector uses the ZENworks Web Server to process the XML-RPC requests.

5. The Selector validates the inventory information and places the information in the Inventory database.
6. The Storer updates the database with the inventory information.
7. The ZENworks administrator views the inventory information.

Deploying Inventory in Multiple or Enterprise Sites

The following sections contain detailed information to help you deploy Server Inventory in multiple or enterprise sites:

Designing the Inventory Tree

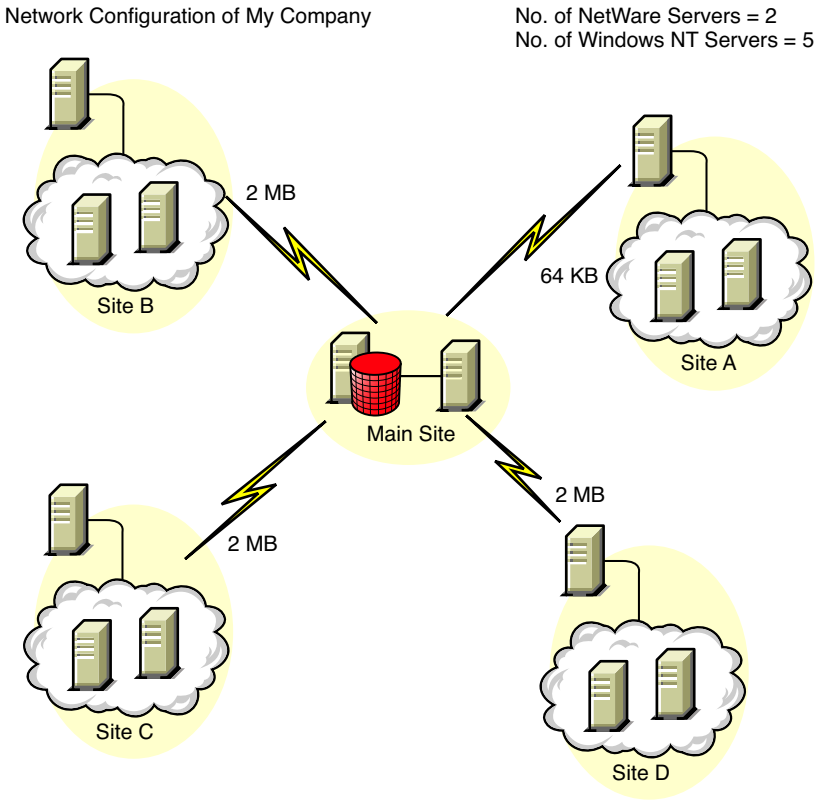
In an enterprise or multiple site, complete the following tasks, in order, to design the inventory tree:

1. List the sites in the enterprise

Describe the entire network of your enterprise.

- List the various sites in your enterprise.
- List the physical links between the various sites.
- Identify the type of links in terms of bandwidth and reliability.

The following figure illustrates the network organization of a enterprise with servers in different locations.



This illustration depicts four sites (Site A, Site, B, Site C, and Site D) connected to a central site. It depicts the physical links between the sites and the type of links in terms of bandwidth.

2. What is the ideal place for the Root Server?

The Root Server in the inventory tree is the highest-level server. Necessarily, an Inventory database is attached to the Root Server.

The inventory information available from the Inventory database of the Root Server will consist of all information from lower-level sites on the network and from the Root Server site.

Factors that you must consider include:

- There must be high-speed links between the Root Server and the management console.
- We recommend that there should be high-speed links between the site having the Root Server and the sites having the lower-level Inventory servers.
- Using the management console, the administrator can collect the inventory information from any of the sites connected on high-speed links from the Root Server, or from the Root Server level site.
- A database server of suitable configuration should be provided for the Inventory server.

3. Is any other database needed?

In addition to the database at the Root Server, you can maintain database servers at different sites.

You might want to maintain additional databases if there are sites or subtrees that are managed for inventory at different locations, and these sites are connected to the network over a slow link.

You should also determine if there are specific reasons to have a separate database for a single site or a set of sites. There might be some organizational needs for your enterprise to have the database server on different sites, even if there is no product deployment need to have any other database.

NOTE: For a majority of enterprises, there could be no need to have any other database besides the enterprise-wide single database.

Optional step: If another database is needed

- If you decide to have additional database servers, identify the sites that need a database. Additionally, you need to examine whether the database will cater to the local site or a site with many subsites. Also, identify the sites that require information in each Inventory database.
- All the sites served by a single database should typically access this database instead of the database at the Root Server for inventory management. This reduces the load on the database at Root Server.
- Database administrators should be available for these sites.

4. Identify the route for Inventory information

Identify the routes for inventory information for all Inventory servers to the nearest database.

To devise a route plan:

- Each route can have an Intermediate Server at a staging site. The Intermediate Server receives and transmits the information to the next destination. These are application-layer-level routes for inventory information. There can be various network-layer-level routes between two adjacent servers, which will be determined and managed by the routers in the network.
- The route provides information indicating how inventory information travels from a particular site to its final destination, which is the database at the Root Server.
- There can be multiple routes. Choose the fastest and most reliable route. To determine the route, consider the physical network links.
- Routes identified and made operational can be changed later, although there might be some cost in terms of management and traffic generation. If there is no intermediate database involved, you can change the route by changing the eDirectory-based policy.
- Put Intermediate Servers on sites where the link parameters change substantially. Criteria to consider are difference in bandwidth, difference in reliability of the links, and the need for roll up of inventory information.
- Availability of Inventory servers on the intermediate site for staging the inventory information should be considered while deciding the sites for Intermediate Servers. Ensure that there is enough disk space on these servers to store all the inventory information on the disk until the Sender sends it to the next destination.

5. Identify servers on each site to act as Inventory and Database Servers

In ZENworks 6.5 Server Management, you choose the role for each Inventory server. For more information, see “Understanding the Inventory Server Roles” on page 420.

The number of inventoried servers attached to an Inventory server also determines the load. The following table lists the disk space requirements for the server:

Server Type	Disk Space Requirements
Leaf Server	(*n1 x s*) + (*n1 x z*)
Leaf Server with Database	(*n1 x s* x 2) + {*(n1 x dbg*)}
Intermediate Server	*n2 x z*
Intermediate Server with Database	(*n2 x z)* + (*n2 x s*) + {(*n2 x dbg)}*
Intermediate Server with Inventoried Servers	(*n1 x s x 2)* + (*n2 x z*)
Intermediate Server with Database and Inventoried Servers	(*n1 x s x 2)* + (*n2 x z*) + *(n2 x s*) + {(*n1 x dbg)* + (*n2 x dbg*)}
Root Server	(*n2 x z)* + *(n2 x s*) + {(*n2 x dbg*)}
Root Server with Inventoried Servers	(*n1 x s x 2)* + (*n2 x z*) + *(n2 x s*) + {(*n1 x dbg)* + (*n2 x dbg*)}
Standalone Server	(*n1 x s x 1)* + {(*n1 x dbg*)}

In the table, *n1* is the number of inventoried servers attached to the server.

s is the size of the scan data files. This file size varies depending on the information collected. Calculate 400 KB scan information from each inventoried server to calculate the load.

dbg is the storage space of the inventory information in the database.

n2 is the number of inventoried servers rolled up to the Inventory server.

z is the size of the compressed scan data file per inventoried server. he average compression ratio is 80-90% of the STR file size.

{ } denotes the disk space of the database server, depending on whether the database is on the same Inventory server or if it is connected to the Inventory server. If the database is on the same Inventory server, calculate the total disk space including the database space for the Inventory server. For example, if the Leaf Server with Database has the Inventory database on the same server, calculate the requirements for storage of inventory information, including the database disk space.

6. Create the tree of servers for enterprise Inventory collection

Ensure that the inventory tree you design follows these guidelines:

- The root of the tree is the Root Server.
- At least one Inventory server per site is recommended.
- Each site has inventoried servers to be scanned.
- Optionally, there will be databases and Intermediate Servers on different sites.

7. Create an implementation plan

After you design the inventory tree, you should develop an implementation plan to cover the phased deployment plan for the network. Use the top-down deployment of the Server Inventory installation. Always begin the installation at the topmost level server (Root Server) and proceed with the next lower-level servers.

8. Start the actual deployment

After your implementation plan is finalized, start the actual deployment according to the plan.

Follow these steps:

1. Install the Inventory servers on the sites. For more information, see “Installing Server Inventory” on page 449.
2. Create and configure the policies applicable to Inventory server and inventoried servers. For more information, see “Configuring the Required Policies” on page 449.

Adding a Database Server to an Existing Inventory Setup

If you have already configured the servers for inventory setup, and you need to add another database server, follow these instructions:

1. Run the installation program to install the Inventory database on the server.

   The installation program installs the Sybase database. If you are maintaining the database in Oracle, make sure that the Oracle database exists. For more information, see “Setting Up the Oracle Inventory Database” on page 471. If you are maintaining the database in MS SQL, make sure that the MS SQL database exists. For more information, see “Setting Up the MS SQL Server 2000 Inventory Database” on page 478

2. Shut down the Inventory services. For more information, see “Stopping the Inventory Service on a NetWare Inventory Server” on page 454.

3. Based on the database you select, make sure that you configure the database. For more information, see “Configuring the Database Location Policy” on page 485.

4. If you want to attach a new database to an existing Inventory server, which does not have a database attached, you must change the role of the Inventory server in its Inventory Service object (Inventory Service_*server_name*). For example, if you attach an Inventory database to an existing Leaf Server, you must change the server role from Leaf Server to Leaf Server with Database in the Inventory Service object of the Leaf Server.

   If you want to attach an Inventory database to a freshly installed Inventory server, you must choose an appropriate server role for the Inventory server.

   To change the role of an Inventory server:

   4a. In ConsoleOne, right-click the Inventory Service object, click Properties, then click the Inventory Service Object Properties tab.

   4b. Choose the new role of the Inventory Service object, then click Apply.

   Follow the actions that you need to change the role. For more information, see “Changing the Role of the Inventory Server” on page 455.

5 Make sure that you enforce Full Scan for the Inventory Service object.

5a In ConsoleOne, right-click the Inventory Service object, click Properties, then click the Inventory Service Object Properties tab.

5b Select the Enable Scan option, then click OK.

6 Bring up the Inventory service. For more information, see “Starting the Inventory Service on a NetWare Inventory Server” on page 454.

Deployment Options for Inventory Server and Inventory Database

The following sections cover these scenarios:

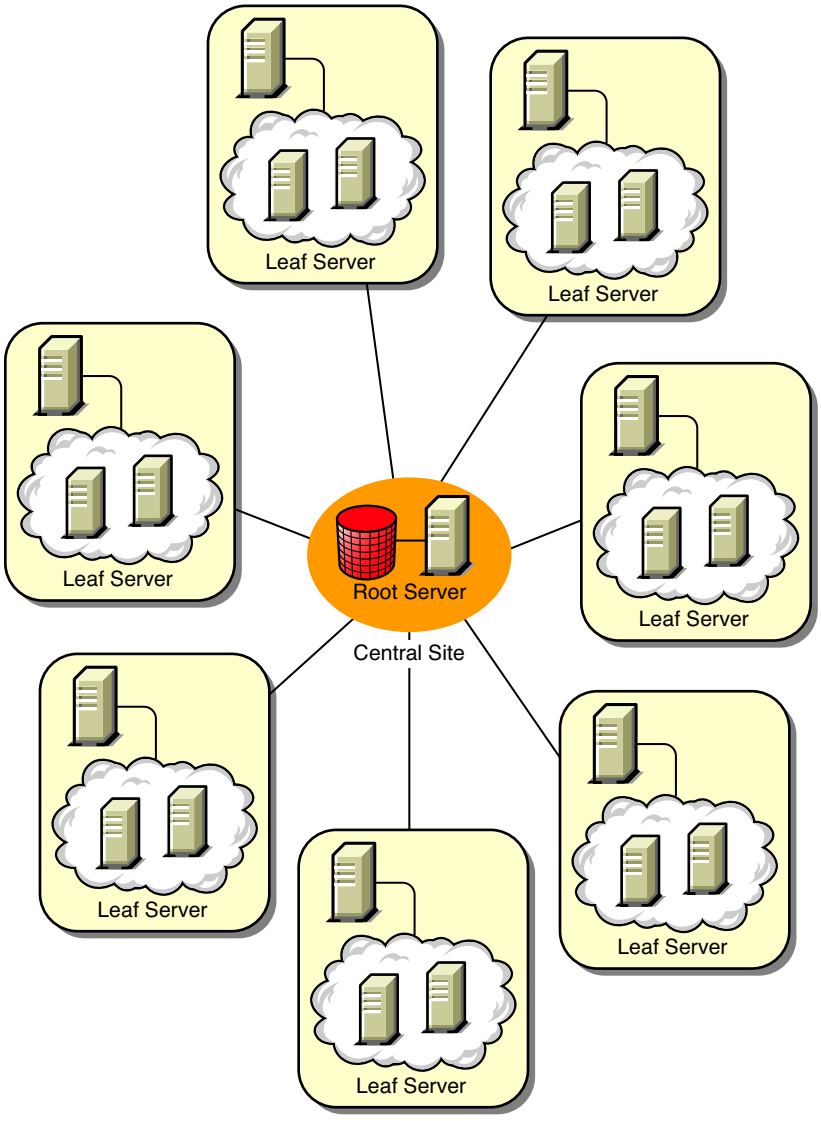
- “Scenario 1: Inventory Deployment without Intermediate Servers in WAN” on page 443
- “Scenario 2: Inventory Deployment with Intermediate Servers in a WAN” on page 445
- “Scenario 3: Roll-Up of the Inventory information Across eDirectory Trees” on page 446
- “Scenario 4: Merging eDirectory Trees” on page 447
- “Scenario 5: Deploying Inventory Server Across Firewall” on page 447

Scenario 1: Inventory Deployment without Intermediate Servers in WAN

In this scenario, the network consists of many remote sites connected to a Central Site over WAN. Each remote site has a Leaf Server that collects inventory information from inventoried servers located in the same site, and rolls up the inventory information to the Root Server located at the central site. The remote sites are administered from the Central Site because the Leaf Servers do have Inventory database attached to it.

TIP: To locally administer the remote sites, you must have Inventory database attached to Leaf Servers and change the role of the Inventory server to Leaf Server with Database. For more information on how to change the role of an Inventory server, see “Changing the Role of the Inventory Server” on page 455.

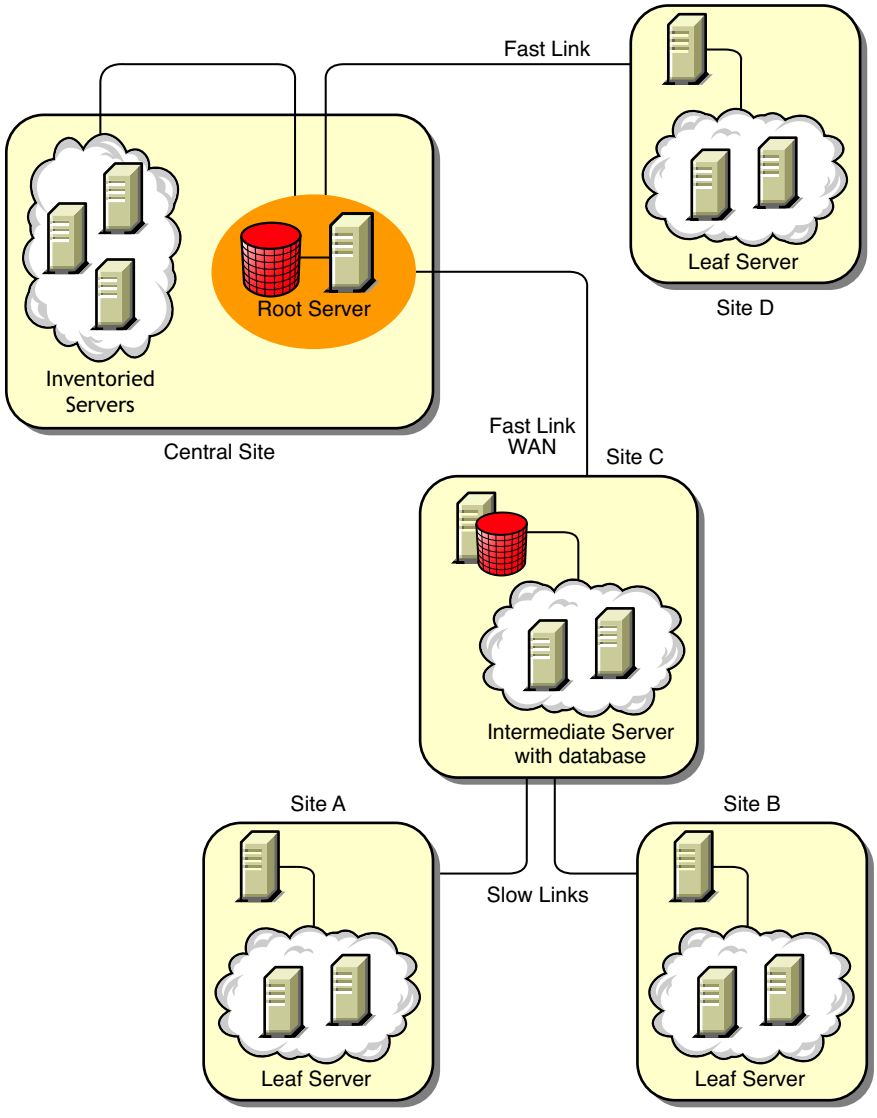
This scenario is illustrated in the following figure.



Scenario 2: Inventory Deployment with Intermediate Servers in a WAN

In this scenario, the network consists of four remote sites (A, B, C, and D), and a Central Site. Sites A and B are connected to Site C over slow links and are not directly connected to the Central Site. Site C is connected to the Central Site over a fast WAN link. Site D is directly connected to the Central Site over a fast link. Sites A, B and C are administered at Site C.

This scenario is illustrated in the following figure:



To administer the enterprise from the Central Site, and also administer Sites A and B from Site C, do the following:

1. Install Leaf Servers at Sites A, B, and D.
2. Install Intermediate Servers with Database at Site C.
3. Configure Leaf Servers at Sites A and B to roll up the inventory information to the Intermediate Server with Database at Site C.
4. Configure the Intermediate Server with Database at Site C to roll up the inventory information to Root Server at the Central Site.

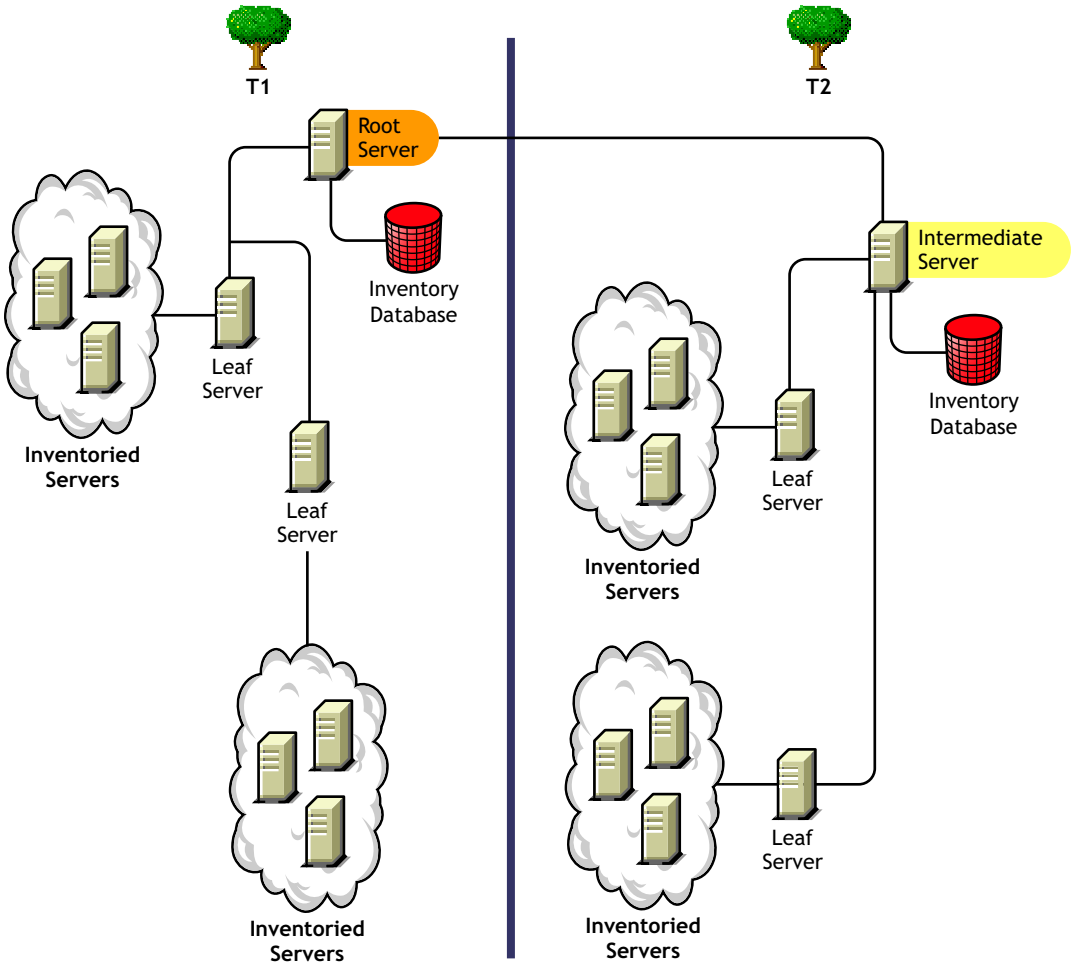
5. Configure the Leaf Server at Site D to roll up the inventory information to Root Server at the Central Site.

Scenario 3: Roll-Up of the Inventory information Across eDirectory Trees

In this configuration, you can deploy any of the previous scenarios. The highest-level Inventory server of one eDirectory tree rolls up the inventory information to an Inventory server located on the other eDirectory tree.

In this configuration, you must install the Distributor on each eDirectory tree for the policies to be distributed.

The following illustration depicts a sample scenario where you can deploy this inventory configuration.



There are two organizations: A and B. Each organization has its own eDirectory tree and inventory tree. Organization A has two Leaf Servers and a Root Server in its inventory tree. Organization B also has two Leaf Servers and a Root Server in its inventory tree. A decision is taken to merge both the organizations and both the inventory trees but to retain the eDirectory trees. After the merger, the role of the Root Server on the eDirectory tree T2 is changed to Intermediate Server with Database and the inventory information is rolled up from the Intermediate Server to the Root Server residing on the eDirectory tree T1.

Scenario 4: Merging eDirectory Trees

In this configuration, you can merge the inventory trees and the eDirectory trees. After you merge the eDirectory trees, you must manually change the eDirectory tree name and (optionally) the Inventory Service DN in the *inventory_server_installation_drive or volume*\zenworks\inv\server\wminv\properties\config.properties file before starting the Inventory service. For more information on merging the eDirectory trees, see the Novell eDirectory documentation Web site (http://www.novell.com/documentation).

To merge the inventory trees, you must change the role of the Root Server of one inventory tree to roll up to an Inventory server in the other inventory tree.

To change the eDirectory tree name and the DN of an Inventory server, edit the following entries of the config.properties file:

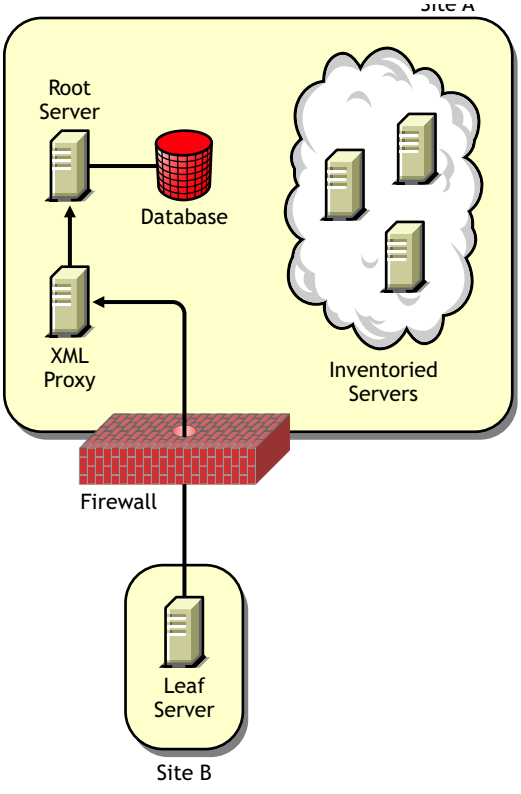
`NDSTree=Target_eDirectory_tree_name`

`InventoryServiceDN=New_DN_of_the_Inventory_server`

Scenario 5: Deploying Inventory Server Across Firewall

There are two sites; Site A and Site B connected through a WAN link. The Inventory server of Site A rolls up to an Inventory server in Site B. All communication from Site A to Site B flows through the firewall at Site B.

The following illustration depicts a sample scenario where you can deploy this inventory configuration:



To enable the roll-up:

- Install an XML proxy at Site A. For more information about installing the Proxy, see "Installation on NetWare and Windows" in "Policy-Enabled Server Management Installation" in the *Novell ZENworks 6.5 Server Management Installation Guide*.
- You must have at least one XML proxy/site installed. One proxy server can handle requests for multiple Inventory servers.
- You can configure the port that the proxy listens to during the ZENworks 6.5 Server Management installation. For more information, see "Installation on NetWare and Windows" in "Policy-Enabled Server Management Installation" in the *Novell ZENworks 6.5 Server Management Installation Guide*.

  You must allow requests to the proxy server on this port at the firewall. You can configure the XML proxy to listen to standard ports allowed by your firewall.

  The XML proxy does not support any commercial Web server. You must ensure that the port number assigned to the XML proxy is not used by any other service on the same server.

  You must also configure the Roll-Up policy with the XML proxy server's address and port number.

Recommendations for Deployment

- When you configure the inventory scanning of inventoried servers, we recommend staggering the inventory scanning to scan at different times or to scan some inventoried servers at a time.
- If many inventoried servers are attached to the same inventory server, we recommend that you do not schedule the scan of all inventoried servers at the same time, because this will stress the Novell eDirectory and the Inventory service.
- You can attach inventoried servers to the server as determined by the number of connections supported by NetWare or Windows servers up to a maximum of 5,000 inventoried servers.
- When you schedule the roll-up of information in the Inventory policies, we recommend the roll-up frequency should be at least one day. If the roll-up of inventory information is scheduled too frequently, for example less than one hour, there may be some performance degradation of the inventory server.
- Use top-down deployment for Inventory installation. Always begin the installation at the topmost level server and proceed with the next lower-level servers. For example, in an inventory setup with a Root Server and a Leaf Server, complete the inventory installation at the Root Server, and then run the installation for the Leaf Server.
- If an Inventory server must receive server Inventory scans either directly from the Inventory servers or through roll-up, you must install ZENworks 6.5 Server Management on this server.
- We recommend that you configure the DNS for your Inventory and database servers. If you have not configured the DNS, choose the IP address in the roll up and database location policies. Scheduling the frequency of information gathering and roll-up must be fine-tuned based on the Root Server. Make sure that the Root server is able to handle the load of the .STR files.

Installing Server Inventory

For detail information on installing Server Inventory, see "Installation on NetWare and Windows" in "Policy-Enabled Server Management Installation" in the *Novell ZENworks 6.5 Server Management Installation Guide*.

Understanding the Effects of Server Inventory Installation

For detail information on the effects of Server Inventory installation, see "Understanding the Effects of Server Inventory Installation" on page 452.

Configuring the Required Policies

The following table lists the actions that you should follow to setup the server for Server Inventory.

To set up this type of server:	Do this:
Standalone Server	1. Follow the steps in "Configuring the Database Location Policy" on page 485 2. Follow the steps in "Configuring the Server Inventory Policy" on page 486
Root Server	1. Follow the steps in "Configuring the Inventory Service Object" on page 484 2. Follow the steps in "Configuring the Database Location Policy" on page 485
Root Server with Inventoried Servers	1. Follow the steps in "Configuring the Inventory Service Object" on page 484. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 486. 3. Follow the steps in "Configuring the Database Location Policy" on page 485.
Intermediate Server	1. Follow the steps in "Configuring the Inventory Service Object" on page 484. 2. Follow the steps in "Configuring the Roll-Up Policy" on page 488.
Intermediate Server with Database	1. Follow the steps in "Configuring the Inventory Service Object" on page 484. 2. Follow the steps in "Configuring the Roll-Up Policy" on page 488. 3. Follow the steps in "Configuring the Database Location Policy" on page 485.
Intermediate Server with Inventoried Servers	1. Follow the steps in "Configuring the Inventory Service Object" on page 484. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 486. 3. Follow the steps in "Configuring the Roll-Up Policy" on page 488.
Intermediate Server with Database and Inventoried Servers	1. Follow the steps in "Configuring the Inventory Service Object" on page 484. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 486. 3. Follow the steps in "Configuring the Roll-Up Policy" on page 488. 4. Follow the steps in "Configuring the Database Location Policy" on page 485.
Leaf Server	1. Follow the steps in "Configuring the Inventory Service Object" on page 484. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 486. 3. Follow the steps in "Configuring the Roll-Up Policy" on page 488.

To set up this type of server:	Do this:
Leaf Server with Database	1. Follow the steps in “Configuring the Inventory Service Object” on page 484. 2. Follow the steps in “Configuring the Server Inventory Policy” on page 486. 3. Follow the steps in “Configuring the Roll-Up Policy” on page 488. 4. Follow the steps in “Configuring the Database Location Policy” on page 485.

Starting the Inventory Service

After installing ZENworks 6.5 Server Management, the Inventory service is automatically started only if you have configured the Inventory Standalone Configuration settings during the installation.

To manually start the Inventory service, see “Starting the Inventory Service on a NetWare Inventory Server” on page 454.

Updating the Software Dictionary

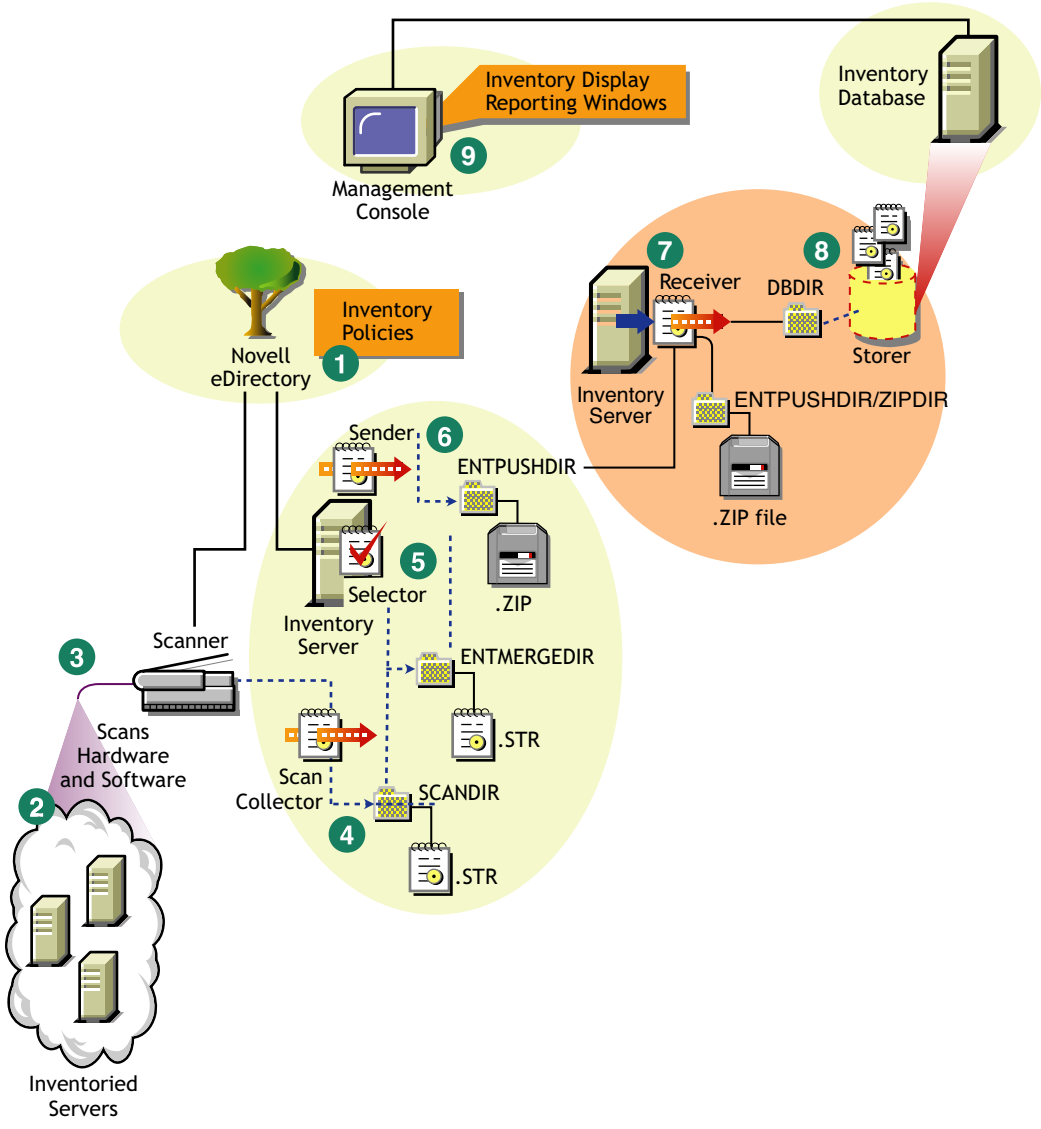
You can update the Software Dictionary in any one of the following ways:

- On each Inventory server, manually download the latest version of the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) and update the software dictionary.
- Manually download the latest version of the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) on an Inventory server (preferably, the Root Server) and automatically distribute the dictionary to all servers in your setup by configuring the “Configuring the Dictionary Update Policy” on page 489. For more information, see “Setting Up Distribution of Dictionary” on page 490.

NOTE: The dictionary is updated and published once every three months in this TID.

Understanding Rolling Up Inventory Information Across Servers

The following illustration depicts rolling up the inventory information across servers, which is explained below:



If the inventory deployment rolls up inventory information across servers, the process of scanning is as follows:

1. The inventory policies in eDirectory define the inventory settings, such as the Inventory Service object name of the Inventory server to which the inventory information will be sent, scanning time, and the software rules for software scan. These settings are customizable.
2. The Scanner uses Policy and Distribution Services to read the inventory policies and collects the inventory information based on the policy settings. The Inventory scanner also checks whether an updated dictionary is available at its Inventory server and downloads the updated dictionary
3. The Scanner stores the inventory information locally on the inventoried server. This information is transferred to the Inventory server using the XML-RPC protocol.

4. The Scan Collector receives the inventory information using the XML-RPC protocol and stores the .str file in the scan directory at the Inventory server. The Scan Collector uses the ZENworks Web Server to process the XML-RPC requests.
5. The Selector validates the inventory information and places it in the enterprise merge directory for roll-up of inventory information. If there is a database attached, the Selector also places the files in the database directory.
6. The Sender on the Inventory server has a Roll-Up policy to identify the Inventory server to which it will transmit the inventory information and the Roll-Up schedule specifies the time for roll-up of information. The Sender compresses the .str files as a .zip file and places the .zip file in the enterprise push directory. The Sender then sends the .zip file to the Receiver on the next-level Inventory server.
7. The Receiver on the next-level Inventory server receives the .zip file.

   NOTE: The next-level Inventory server can be located on the same eDirectory tree or on a different eDirectory tree.

   On the Intermediate Server, the Receiver copies the file in the enterprise push directory (entpushdir). On the Intermediate Server with Database, or the Intermediate Server with Database and Inventoried Servers, the Receiver places the file in \entpushdir and places the file to the database directory (dbdir).

   On the Root Server, or the Root Server with Inventoried Servers, the Receiver copies the file to the \dbdir only.
8. The Storer extracts the .zip file containing the .str files to a temp directory (\dbdir\temp) and updates the database with the inventory information of the inventoried server .str file.
9. The ZENworks administrator views the inventory information, and queries the database in ConsoleOne.

Understanding the Effects of Server Inventory Installation

On the Inventory server, the ZENworks 6.5 Server Inventory installation program does the following:

- On a NetWare Inventory server:
  - Copies the inventory related files to the *installation_volume*.
  - Copies the Server Inventory snap-ins to the ConsoleOne directory.
  - Creates an Inventory Service object in eDirectory for each server on which the Inventory server is installed. This object is populated with the following attributes: zeninvRole (role of the server), zeninvScanFilePath (path to \scandir directory) zeninvHostServer (DN of the server on which Inventory server is installed), and zeninvDictionarypath (Path to the dictionary directory).
  - If the Inventory Service object already exists, the object is validated and re-created if it is invalid.
  - During installation, the Inventory Service object is made a trustee of the NCP™ server with Compare and Read rights.
  - The installation program assigns the Inventory Service object as trustee to itself.
  - Creates the scan directory with the subdirectories in the specified volume on the Inventory server. [Root] is granted the Create rights to this directory.

- Creates a dictionary directory (dictdir), and copies the files of general dictionary and private dictionary. [Root] is granted the Read and Write rights to this directory.
- Creates the zenworks.properties file in sys:\system. This file contains the installation path of the Inventory server and the ZENworks Web Server.
- During the Server Inventory installation, if you have configured Inventory Standalone Configuration settings, then the Inventory Service Manager is automatically started.
- Installs the ZENworks Web Server on the Inventory server, if not installed previously.
- If Server Inventory is reinstalled in the same directory as the previous installation, the config.properties and directory.properties files are backed up and re-created.

- On a Windows Inventory server:
  - Copies the inventory related files to the *installation_directory*.
  - Copies the Server Inventory snap-in component to the ConsoleOne directory.
  - Creates the scan directory with the subdirectories in the specified volume on the Inventory server, and creates a share with Create rights to this directory for all users.
  - Creates a dictionary directory (dictdir), copies the files of general dictionary and private dictionary, and grants Read and Write rights to this directory for all users.
  - Creates an Inventory Service object in eDirectory for each server on which the Inventory server is installed. The following attributes are populated: zeninvRole (role of the server), zeninvScanFilePath (path to \scandir), zeninvHostServer (DN of the server on which Inventory is installed), and zeninvDictionarypath (path to the dictionary directory).
  - If the Inventory Service object already exists, the object is validated and re-created if it is invalid.
  - During installation, the Inventory Service object is made a trustee of the NCP server with Compare and Read rights.
  - The installation program assigns the Inventory Service object as trustee to itself.
  - The Inventory Service Manager is created as a service.
  - Edits the Registry settings to add the installation path of the Inventory server and the ZENworks Web Server.
  - On the Inventory server, the ZENworks Service Management is created as a service.
  - If Server Inventory is reinstalled in the same directory as the previous installation directory, the config.properties and directory.properties files are backed up and re-created.

On the database servers:

- Installs the Sybase database on the server you specify.
- If the database server is installed in the previous installation directory, the database files are re-created if they were found invalid or non-existing.
- If Sybase is already installed, only the database files are copied.
- On NetWare, the mgmtdb.db entries are added to the sys:\system\mgmtdbs.ncf file. On Windows, the mgmtdb.db entries are added to the registry.
- Creates a database object (Inventory database_*server_name*) for Sybase and configures the properties of the object.

- At server startup time, the database is loaded.

Starting and Stopping the Inventory Service

The section provides information on:

- “Starting the Inventory Service on a NetWare Inventory Server” on page 454
- “Stopping the Inventory Service on a NetWare Inventory Server” on page 454
- “Starting the Inventory Service on a Windows Inventory Server” on page 454
- “Stopping the Inventory Service on a Windows Inventory Server” on page 455

For more information about the various Inventory services, see “Understanding the Inventory Service Manager” on page 493.

Starting the Inventory Service on a NetWare Inventory Server

Before you start the Inventory service, make sure that the Inventory database are up and running. The Inventory database will be automatically started after the installation.

To start the Inventory services on the NetWare Inventory server, enter **`startinv`** at the server console prompt.

To start an Inventory service, enter **`startser` *`inventory_service_name`*** at the server console prompt.

After starting the Inventory service, make sure that the Inventory services are up and running. To list all services, enter **`listser *`** at the server console prompt. To list an Inventory service, enter **`listser` *`inventory_service_name`*** at the server console prompt.

If the services are not up and running, check the Server Status log. For more information on the Server Status log, see “Viewing the Status of Inventory Components on an Inventory Server” on page 678.

Stopping the Inventory Service on a NetWare Inventory Server

To stop an Inventory service, enter **`stopser` *`Inventory_service_name`*** at the server console prompt.

To stop all the Inventory services, enter **`stopser * at the server console prompt`**.

Starting the Inventory Service on a Windows Inventory Server

Before you start the Inventory service, make sure that the ZENworks Server Management components and the Inventory database are up and running. The Inventory database will be automatically started after the installation.

To start the Inventory services on the Windows 2000/2003 Inventory server:

1. In the Control Panel, double-click Administrative Tools.
2. Double-click Services.
3. Select Novell Inventory Service, then click Start.

To start an Inventory service from the console prompt:

1. Go to the *installation_directory*\inv\server\wminv\bin directory.
2. At the prompt, enter **`startser inventory_service_name`**.

After starting the Inventory service, make sure that the Inventory services are up and running. To list all services, enter **`listser "*"`** at the server console prompt. To list an Inventory service from the console prompt:

1. Go to the *installation_directory*\inv\server\wminv\bin directory.
2. At the prompt, enter **`listser inventory_service_name`**.

If the services are not up and running, check the Server Status log. For more information on the Server Status log, see "Viewing the Status of Inventory Components on an Inventory Server" on page 678.

Stopping the Inventory Service on a Windows Inventory Server

To stop the Inventory services on a Windows 2000/2003 Inventory server:

1. In the Control Panel, double-click Administrative Tools.
2. Double-click Services.
3. Select Novell Inventory Service, then click Stop.

To stop a service on a Windows Inventory server from the console prompt:

1. Go to the *installation_directory*\inv\server\wminv\bin directory.
2. Enter **`stopser inventory_service_name`**.

Changing the Role of the Inventory Server

When you install ZENworks 6.5 Server Management, by default, the role of the Inventory server is a Standalone Server. By configuring the Inventory Service object, you can assign specific roles to the Inventory server based on your inventory deployment.

For example, if the deployment plan identifies three Inventory servers, such as a Root Server, an Intermediate Server with Database, and a Leaf Server for inventory deployment, you install Server Inventory on these servers, and choose the role for the Inventory server. Later, if you want to make changes in the inventory deployment, such as attaching the inventoried servers to the existing Root Server, you need to change the role of the Inventory Service object from Root Server to Root Server with Inventoried Servers. Additionally, depending on the new role, there are some policies you need to configure.

To change the role for any Inventory server:

1. Plan the change of roles carefully because the changes will impact the existing inventory deployment. Also, consider the disk space requirements and ensure that you have the required configurations for Inventory.
2. In ConsoleOne, right-click the Inventory Service object (Inventory Service_*server_name*), click Properties, then click the Inventory Service Object Properties tab.
3. Choose the new role of the Inventory Service object, then click Apply.
4. Bring down the services running on the changed Inventory server, follow the actions that you need to change the role, and then bring up the Inventory services.

To stop all Inventory Services:

- At NetWare server console prompt, enter the following commands:

  ```
  stopser *
  ```

  ```
  java -killZenWSInv
  ```

- On the Windows 2000/2003 server, from the Services window, click Novell Inventory Services, then click Stop.

To restart all Inventory Services:

- At NetWare server console prompt, enter **`startinv`**
- On the Windows 2000/2003 server, from the Services window, click Novell Inventory Services, then click Start.

The following sections contain information to help you change the role of the Inventory Service object:

- "Changing the Role of the Root Server" on page 456
- "Changing the Role of the Root Server with Inventoried Servers" on page 457
- "Changing the Role of the Intermediate Server" on page 458
- "Changing the Role of the Intermediate Server with Database" on page 459
- "Changing the Role of the Intermediate Server with Database and Inventoried Servers" on page 460
- "Changing the Role of the Intermediate Server with Inventoried Servers" on page 461
- "Changing the Role of the Leaf Server" on page 462
- "Changing the Role of the Leaf Server with Database" on page 463
- "Changing the Role of the Standalone Server" on page 464

Changing the Role of the Root Server

To change the role of the Root Server to a different role, follow the actions specified in the following table:

To change the role of the Root Server to …	Tasks:
Root Server with Inventoried Servers	Perform the following task: 1. After changing the role, configure the Server Inventory policy so that the inventoried servers that you have attached to the Root Server with Inventoried servers will be scanned for.
Intermediate Server	Perform the following tasks: 1. Before changing the role, remove the Database Location policy associated with a Root Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from this Inventory server.

To change the role of the Root Server to …	Tasks:
Intermediate Server with Database	Perform the following task: 1. After changing the role, configure the Roll-Up policy to specify the next-destination Inventory server for roll-up of information from this Inventory server.
Intermediate Server with Database and Inventoried Servers	Perform the following tasks after changing the role: 1. Configure the Server Inventory policy so that the inventoried servers that you have attached will be scanned for. 2. Configure the Roll-Up policy to specify the next-destination server for roll-up of information from this Inventory server.
Intermediate Server with Inventoried Servers	Perform the following tasks: 1. Before changing the role, remove the Database Location policy associated with the Root Server. 2. After changing the role, configure the Server Inventory policy so that the inventoried servers that you have attached will be scanned for. 3. After changing the role, configure the Roll-Up policy to specify the next-destination Inventory server for roll-up of information from this Inventory server.
Leaf Server, Leaf Server with Database, or Standalone Server	Server Inventory does not allow you to change the Root Server to these Inventory servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Server Inventory component.

Changing the Role of the Root Server with Inventoried Servers

Follow the actions specified in the following table:

To change the role of the Root Server with Inventoried Servers to …	Tasks:
Root Server	Perform the following task: 1. Before changing the role, remove the Server Inventory policy associated with the Root Server with Inventoried Servers.
Intermediate Server	Perform the following tasks: 1. Before changing this role, remove the Database Location policy and the Server Inventory policy. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from this Inventory server.
Intermediate Server with Database	Perform the following tasks: 1. Before changing the role, if the Server Inventory policy is associated with the Root Server with Inventory servers, remove the policy for those servers attached to this Inventory server or to the lower-level Inventory servers that roll up to this Inventory server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from this Inventory server.

To change the role of the Root Server with Inventoried Servers to …	Tasks:
Intermediate Server with Database and Inventoried Servers	Perform the following task: 1. After changing the role, configure the Roll-Up policy to specify the next-destination Inventory server for roll-up of information from this Inventory server.
Intermediate Server with Inventoried Servers	Perform the following task: 1. Before changing the role, remove the Database Location policy that is associated with the Root Server with Inventoried Servers.
Leaf Server, Leaf Server with Database, or Standalone Server	Server Inventory does not allow you to change the Root Server to these Inventory servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Server Inventory component.

Changing the Role of the Intermediate Server

Follow the actions specified in the following table:

To change the role of the Intermediate Server to …	Tasks:
Root Server	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy. 2. After changing the role, configure the Database Location policy.
Root Server with Inventory Servers	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy. 2. After changing the role, configure the Server Inventory policy for those inventoried servers attached to this server and the Database Location policy.
Intermediate Server with Database	Perform the following task: 1. After changing the role, configure the Database Location policy for this Inventory server.
Intermediate Server with Database and Inventoried Servers	Perform the following tasks: 1. After changing the role, configure the Server Inventory policy so that all the inventoried servers associated to this Inventory Service object, and also those inventoried servers associated to the lower-level Inventory servers that roll up to this Inventory server will be scanned for. 2. After changing the role, configure the Database Location policy.
Intermediate Server with Inventoried Servers	Perform the following task: 1. After changing the role, configure the Server Inventory policy so that the inventoried servers that you have attached will be scanned for.
Leaf Server, Leaf Server with Database, or Standalone Server	Server Inventory does not allow you to change the Intermediate Server to these Inventory servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Server Inventory component.

Changing the Role of the Intermediate Server with Database

Follow the actions specified in the following table:

To change the role of the Intermediate Server with Database to …	Tasks:
Root Server	Perform the following task: 1. Before changing the role, remove the Roll-Up policy associated with the Intermediate Server with Database.
Root Server with Inventoried Servers	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy associated with the Intermediate Server with Database. 2. After changing the role, configure the Server Inventory policy so that the inventoried servers that you have attached will be scanned for.
Intermediate Server	Perform the following task: 1. Before changing the role, remove the Database Location policy that is associated with the Intermediate Server with Database.
Intermediate Server with Database and Inventoried Servers	Perform the following task: 1. After changing the role, configure the Server Inventory policy so that the inventoried servers attached will be scanned for.
Intermediate Server with Inventoried Servers	Perform the following tasks: 1. Before changing the role, remove the Database Location policy that is associated with the Intermediate Server with Database. 2. After changing the role, configure the Server Inventory policy so that the inventoried servers that you have attached will be scanned for.
Leaf Server, Leaf Server with Database, or Standalone Server	Server Inventory does not allow you to change the Intermediate Server to these Inventory servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Server Inventory component.

Changing the Role of the Intermediate Server with Database and Inventoried Servers

Follow the actions specified in the following table:

To change the role of the Intermediate Server with Database and Inventoried Servers to …	Tasks:
Root Server	Perform the following tasks before changing the role: 1. Remove the Roll-Up policy associated with the Intermediate Server with Database and Inventoried Servers. 2. Remove the Server Inventory policy associated with the inventoried server so that the inventoried servers will not send the scan files to this server.
Root Server with Inventoried Servers	Perform the following task: 1. Before changing the role, remove the Roll-Up policy associated with the Intermediate Server with Database and Inventoried Servers.
Intermediate Server	Perform the following tasks before changing the role: 1. Remove the Server Inventory policy associated with the lower-level servers that roll up to the Intermediate Server with Database and Inventoried Servers. 2. Remove the Database Location policy associated with the Intermediate Server with Database and Inventoried Servers.
Intermediate Server with Database	Perform the following task: 1. Remove the Server Inventory policy of the Intermediate Server with Database and Inventoried Servers or reconfigure the policy.
Intermediate Server with Inventoried Servers	Perform the following task: 1. Before changing the role, remove the Database Location policy associated with the Intermediate Server with Database and Inventoried Servers.
Leaf Server, Leaf Server with Database, Standalone Server	Server Inventory does not allow you to change the Intermediate Server to these servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Server Inventory component.

Changing the Role of the Intermediate Server with Inventoried Servers

Follow the actions specified in the following table:

To change the role of the Intermediate Server with Inventoried Servers to …	Tasks:
Root Server	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy associated with the Intermediate Server with Inventoried Servers. 2. Before changing the role, remove the Server Inventory policy associated with the inventoried server so that the inventoried servers attached will not send the scan files to this Inventory server. 3. After changing the role, configure the Database Location policy for this Inventory server.
Root Server with Inventoried Servers	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy associated with the Intermediate Server with Inventoried Servers. 2. After changing the role, configure the Server Inventory policy for those inventoried servers attached to the lower-level Inventory server that roll up to this Inventory server. 3. After changing the role, configure the Database Location policy.
Intermediate Server	Perform the following task: 1. Before changing the role, remove the Server Inventory policy.
Intermediate Server with Database	Perform the following tasks: 1. Before changing the role, remove the Server Inventory policy associated to the inventoried server attached to this Inventory Service object. 2. After changing the role, configure the Database Location policy for this Inventory server.
Intermediate Server with Database and Inventoried Servers	Perform the following task: 1. After changing the role, configure the Database Location policy for this Inventory server.
Leaf Server, Leaf Server with Database or Standalone Server	Server Inventory does not allow you to change the Intermediate Server to these Inventory servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Server Inventory component.

Changing the Role of the Leaf Server

Follow the actions specified in the following table:

To change the role of the Leaf Server to …	Tasks:
Root Server	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy associated with the Leaf Servers. 2. Before changing the role, remove the Server Inventory policy associated with the inventoried server. 3. After changing the role, configure the Database Location policy for the Root Server.
Root Server with Inventoried Servers	Perform the following tasks: 1. Before changing the role, remove the Roll-Up policy associated with the Leaf Server. 2. After changing the role, configure the Database Location policy for the Root Server with Inventoried Servers.
Intermediate Server	Perform the following tasks: 1. Before changing the role, remove the Server Inventory policy for those inventoried servers associated with the Inventory server or reconfigure the policy.
Intermediate Server with Database	Perform the following tasks: 1. Before changing the role, remove the Server Inventory policy for those inventoried servers associated with the lower-level Inventory servers that roll up to this Inventory server or reconfigure the policy. 2. After changing the role, configure the Database Location policy for this Inventory server.
Intermediate Server with Database and Inventoried Servers	Perform the following task: 1. After changing the role, configure the Database Location policy for this Inventory server.
Intermediate Server with Inventoried Servers	This change of role does not require any specific policy modifications.
Leaf Server with Database	Perform the following task: 1. After changing the role, configure the Database Location policy for this Inventory server.
Standalone Server	Perform the following task: 1. Before changing the role, remove the Roll-Up policy associated with the Leaf Server.

Changing the Role of the Leaf Server with Database

Follow the actions specified in the following table:

To change the role of the Leaf Server with Database to …	Tasks:
Root Server	Perform the following tasks before changing the role: 1. Remove the Server Inventory policy associated with the Leaf Server with Database. 2. Remove the Roll-Up policy associated with the Leaf Server with Database.
Root Server with Inventoried Servers	Perform the following task: 1. Before changing the role, remove the Roll-Up policy associated with the Leaf Server with Database.
Intermediate Server	Perform the following task: 1. Before changing the role, remove the Server Inventory policy and the Database Location policy associated with the Leaf Server with Database.
Intermediate Server with Database	Perform the following task: 1. Before changing the role, remove the Server Inventory policy associated with the Leaf Server with Database.
Intermediate Server with Database and Inventoried Servers	This change of role does not require any specific policy modifications.
Intermediate Server with Inventoried Servers	Perform the following task: 1. Before changing the role, remove the Database Location policy associated with the Leaf Server with Database.
Leaf Server	Perform the following task: 1. Before changing the role, remove the Database Location policy associated with the Leaf Server with Database.
Standalone Server	Perform the following task: 1. Before changing the role, remove the Roll-Up policy associated with the Leaf Server with Database.

Changing the Role of the Standalone Server

Follow the actions specified in the following table:

To change the role of the Standalone Server to …	Tasks:
Root Server	Perform the following task: 1. Before changing the role, remove the Server Inventory policy associated with the Standalone Server.
Root Server with Inventoried Servers	This change of role does not require any specific policy modifications.
Intermediate Server	Perform the following tasks: 1. Before changing the role, remove the Server Inventory policy and the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination Inventory server for roll-up of information from the Intermediate Server with Database.
Intermediate Server with Database	Perform the following tasks: 1. Before changing the role, remove the Server Inventory policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination Inventory server for roll-up of information from the Intermediate Server with Database.
Intermediate Server with Database and Inventoried Servers	Perform the following tasks: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from the Intermediate Server with Database and Inventoried Servers.
Intermediate Server with Inventoried Servers	Perform the following tasks: 1. Before changing the role, remove the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from the Intermediate Server with Inventoried Servers.
Leaf Server	Perform the following tasks: 1. Before changing the role, remove the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from the Leaf Server.
Leaf Server with Database	Perform the following task: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of information from the Leaf Server with Database.

Setting Up the Inventory Database

The following sections contain detailed information to help you set up your Inventory database:

- “Setting Up the Sybase Inventory Database” on page 465
- “Setting Up the Oracle Inventory Database” on page 471
- “Setting Up the MS SQL Server 2000 Inventory Database” on page 478

If you want to replace the Inventory database, always stop the Inventory services before replacing the database. Replace the database and restart the Inventory services. For more information, see “Starting and Stopping the Inventory Service” on page 454.

Setting Up the Sybase Inventory Database

This section contains the following information:

- “Adding Non-English Enumerated Values for Inventory Attributes into the Sybase Inventory Database” on page 465
- “Manually Creating the Sybase Inventory Database Object” on page 466
- “Organizing the Database Spaces for a Sybase Database on NetWare or Windows Servers (AlterDBSpace Tool)” on page 467
- “Understanding the Sybase Database Startup Parameters” on page 469
- “Backing Up the Sybase Inventory Database” on page 469

Adding Non-English Enumerated Values for Inventory Attributes into the Sybase Inventory Database

You need to add the non-English enumerated values so the Inventory report can display the enum value for the inventory attributes in internationalized versions. The non-English enum values must be available in English version of the product so that the rolled-up inventory information from non-English sites can be properly captured at the high-level servers where only English versions are installed.

For more information about the list of attributes that contains enumerated values, see Appendix K, “Enumeration Values,” on page 723.

To add the non-English enum values:

1. Specify the JDBC connection settings in the *zenworks_directory*\inv\server\wminv\properties\connection.prop file to connect to the Sybase database.

   You can do this by copying the template property settings for Sybase specified in the comments section in the connection.prop file. Specify the IP address, port number, and Database SID in the JDBC URL string that matches your Sybase server configuration.

2. At the server prompt, enter **AddEnums** ***directory_name_containing_connection.prop***

   If your Inventory server is running on a Windows machine, run the above command from *zenworks_directory*\inv\server\wminv\bin.

   After executing the command, the a message indicating that the non-English enums have been successfully inserted is displayed on the console prompt.

Manually Creating the Sybase Inventory Database Object

To manually create the Inventory Database object (Inventory database_*server_name*) for Sybase:

1. In ConsoleOne, right-click in the eDirectory tree where you want to create the database object, click New, click Object, click ZENworks Database, then click OK.
2. Enter a name for the database object, then click OK.
3. Configure the Database server options of the Database object.
   - 3a In ConsoleOne, right-click the database object, click Properties, then click the ZENworks Database tab.
   - 3b Select the database server object using any of the following methods:
     - If eDirectory is installed on the database server: in the Server DN field, browse for and select the Server object for the server where the database is physically installed and running.

       The server's IP address is automatically populated to the Server IP Address or DNS Name drop-down list. If the selected server object has more than one IP address, select the appropriate IP address.

       To clear the value set in the Server DN field, type the IP address of another database server or browse and select another server object.
     - If eDirectory is not installed on the database server, then enter the server's IP address or the DNS name in the Server IP Address or DNS Name field.
   - 3c Type the values for the following options:
     - **Database (Read-Write) Username:** *MW_DBA*
     - **Database (Read-Write) Password:** *novell*
     - **Database (Read Only) Username:** *MW_READER*
     - **Database (Read Only) Password:** *novell*
     - **Database (Write Only) Username:** *MW_UPDATER*
     - **Database (Write Only) Password:** *novell*

     IMPORTANT: All Inventory components use the username and the password configured in the database object. By default, "novell" is the password for all options. But you can change it in the database, and update the same here.
   - 3d Click Apply.
   - 3e To configure the JDBC* Driver properties, click the Jdbc Driver Information tab.
   - 3f Select Sybase, then click Default Settings.

     This populates the fields with default JDBC driver information.

     The database settings for Sybase are:
     - **Driver:** *com.sybase.jdbc.SybDriver*
     - **Protocol:** *jdbc:*
     - **SubProtocol:** *sybase*:
     - **SubName:** *Tds:*
     - **Port:** *2638*

- **Flags:** ?ServiceName=mgmtdb&JCONNECT_VERSION=4
- **Database Service Name:** *the database name specified against the -n Sybase startup parameter while invoking Sybase*.

  NOTE: By default, the value of the -n switch is the IP address of the database server. If you retain this switch value, you must enter the same IP address as the database service name.

3g Click Apply, then click Close.

Organizing the Database Spaces for a Sybase Database on NetWare or Windows Servers (AlterDBSpace Tool)

If a NetWare database server has volumes other than SYS: or a Windows database server has additional hard drives, placing the Sybase database spaces files on separate volumes or drives improves performance while accessing the database.

If you install the Sybase database component of ZENworks 6.5 Server Management, the system database file and the database spaces files are installed in the location on the database server you specify. On loading the Inventory database server, the system database file (mgmtdb.db) is loaded. This mgmtdb.db file references the inventory information in the database spaces files. The database spaces files (mgmtdb1.db, mgmtdb2.db, mgmtdb3.db, mgmtdb4.db, mgmtdb5.db, mgmtdb6.db, mgmtdb7.db, mgmtdb8.db, mgmtdb9.db, mgmtdb10.db, and mgmtdb11.db) contain the inventory information.

The alterdb.props file is installed on the database server in the *inventory_server_installation_directory*\wminv\properties directory. You can modify the sections in the file to specify the location of the database spaces on the volumes or drives.

The contents of the alterdb.props file are as follows:

```
#Database Space Properties

count=11

mgmtdb1=location_of_mgmtdb1

mgmtdb2=location_of_mgmtdb2

mgmtdb3=location_of_mgmtdb3

mgmtdb4=location_of_mgmtdb4

mgmtdb5=location_of_mgmtdb5

mgmtdb6=location_of_mgmtdb6

mgmtdb7=location_of_mgmtdb7

mgmtdb8=location_of_mgmtdb8

mgmtdb9=location_of_mgmtdb9

mgmtdb10=location_of_mgmtdb10

mgmtdb11=location_of_mgmtdb11

.....
```

To organize the database spaces:

1. Ensure that the database is not loaded.
2. Ensure that the Inventory Service Manager is not running on the Inventory server.
3. Manually move the database spaces files on the Inventory server.

   Arrange the database spaces files as follows for better performance:

   - MGMTDB1 and MGMTDB2 in the same location
   - MGMTDB3 and MGMTDB6 in the same location
   - MGMTDB5 and MGMTDB7 in the same location
   - MGMTDB8 and MGMTDB4 in the same location
   - MGMTDB9 and MGMTDB10 in the same location
   - MGMTDB11 in a location

   IMPORTANT: If you move mgmtdb.db to another directory or volume on a NetWare server, update the sys:\system\mgmtdbs.ncf file with the new location of the mgmtdb.db.

   If you move mgmtdb.db to another directory or volume on a Windows server, run the ntdbconfig.exe located in the zenworks\database\dbengine directory. In the NTDBCONFIG dialog box, enter the new path of mgmtdb.db.
4. Modify the location of the eleven database spaces files in the alterdb.props file.

   For example, for NetWare, enter:

   ```
   mgmtdb3=sys:\\zenworks\\inv\\db
   ```

   or for Windows, enter:

   ```
   mgmtdb3=c:\\zenworks\\inv\\db
   ```
5. Load the database, then enter **mgmtdbs** on NetWare servers, or on Windows servers, run the database service.

   Ignore the error messages displayed on the console. These messages are displayed because the database spaces files are not loaded.
6. Ensure that the Database Location policy has been configured.
7. On the Inventory server console, run the AlterDBSpace service, then enter **StartSer AlterDBSpace**.

   On the Inventory server, the AlterDBSpace tool runs as a service.

   You will see a message that the database is adjusted.
8. Exit the database and then load the database.

   Ensure that there are no errors while loading the database. Errors indicate that the specified location of the database spaces files are incorrect or does not exist. Ensure that the path to the database spaces files is correct in the alterdb.props file and repeat the procedure to organize the database spaces files.

   IMPORTANT: If you place the database spaces files in different volumes or drives, the log file should be placed in the same volume or drive as the System database file (mgmtdb.db).

Understanding the Sybase Database Startup Parameters

The startup parameters of the Sybase database are as follows:

- **-c:** Sets the initial memory reserves for caching database pages and other server information. For example, -c 32M reserves 32 MB cache size.
- **-gc:** Sets the maximum length of time in minutes that the database server runs without doing a checkpoint on each database. The default value is 60 minutes. For example, -gc sets the checkpoint time as 120 minutes.
- **-gn:** Sets the number of execution threads to be used in the database server.
- **-m:** Deletes the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server.
- **-n:** Specifies the host name of the database server. For example, -n *IP_address.*
- **-ti:** Disconnects the connections that have not submitted a request for a certain number of minutes. The default is 240 (4 hours). A client machine in the middle of the database transaction locks until the transaction ends or the connection terminates. The -ti option is provided to disconnect inactive connections and to free their locks. For example, specify -ti 400.
- **-x:** Specifies a communication link. For example, `-x tcpip` indicates a TCP/IP link.
- **-ct:** Enables character set translation by converting strings between character sets that represent the same characters but at different values. This is useful when the client machine and the database use different character sets.
- **-gss:** Sets the stack size per internal execution thread in the server.
- ***database_installation_path*:** Specifies the installation path of the Inventory database. For example, c:\zenworks\inv\db\mgmtdb.db.

Backing Up the Sybase Inventory Database

Server Inventory provides provides a utility, Database Backup, to back up the Sybase Inventory database from the server. We recommend that you back up the database on a weekly basis. However, if you are tracking the inventory of servers frequently, increase the frequency of backup.

You can back up the database files and the transaction log to the location relative to the \scandir path.

You can run Database Backup either from the server console or ConsoleOne

This section provides information on the following topics:

- “Running Database Backup from the Server Console” on page 469
- “Running Database Backup from ConsoleOne” on page 470
- “Restoring the Inventory Database” on page 471

Running Database Backup from the Server Console

Before running Database Backup from the server console, fulfill the following prerequisites:

- ❑ You can run Database Backup only on an Inventory server to which you have associated a database server. If you deployed more than one database server, you must run Database Backup for each database server.

- Ensure that the database you have to back up is configured in the Database Location policy. For more information on how to access the Database Location policy, see "Configuring the Database Location Policy" on page 485.
- In the [DBBackup Service] section of the server property file, you must modify the location of the backup destination in the ARGUMENTS parameter. The backup files are relative to the SCANDIR path. For example, if the SCANDIR path is sys:\zenworks\inv\scandir, the database is backed up in sys:\zenworks\inv\database*directory_you_specify*. You must modify the server property file located on the server on which you are running Database Backup. Modify the server property file corresponding to the role of the server. For example, if you are running Database Backup on the Leaf Server with Database, modify the server property file, leaf_db_wks.properties.
- Ensure that the Service Manager is loaded when you run Database Backup.

To run Database Backup from the server console:

1. At the Inventory server console, enter **`StartSer DBBACKUP`**
2. View the status of the backup in the backup log file. The database will be copied to *zenworks_installation_path*\zenworks\inv\db*directory_you_specify*

Database Backup creates a log file, bacstatus.txt, located in the zenworks\inv\database directory on NetWare and Windows 2000 servers. The log records the status of the backup operation. Open this text file to view the status of the backup. This file increases in size for every backup operation. Remove the existing contents of the file if you do not require the details.

Running Database Backup from ConsoleOne

1. In ConsoleOne, click Tools, click ZENworks Inventory, then click Database Backup.

   If you want to back up the latest information in the Inventory database, right-click the database object, click ZENworks Inventory, then click Database Backup
2. Enter the path to the directory where the database backup will be saved

   WARNING: Do not use double-byte characters in the directory name. If you do so, Sybase will interpret the double-byte characters as a different name, and will back up the database in the directory with the interpreted name.

   If the Inventory database is running on a NetWare server, you can either enter the path or click Browse to browse for and select a directory. If you enter the database backup directory name without specifying the complete path, the backup directory will be created in the sys: directory.

   If the Inventory database is running on a Windows machine, you must manually enter the backup directory path. If you enter the database backup directory name without specifying the complete path, the backup directory will be created in the \winnt\system32 directory.

   NOTE: If you want to back up the database to a non-existent directory, only one level of the new directory will be created. To back up the database to a subdirectory, ensure that the primary directory already exists. For example, if you want to back up the database to a new c:\backup directory, the \backup directory will be created and the database will be backed up. But if you want to back up the database to a new \database directory, located under c:\backup, the \backup directory must already exist.
3. Click Start Backup

   This backs up the database to the specified directory on the server running the database and overwrites any existing files without prompting about the overwrite.

Database Backup creates a log file, bacstatus.txt, located in the *ConsoleOne_installation_directory*\1.2\bin directory on NetWare and Windows 2000 servers. The log records the status of the backup operation. Open this text file to view the status of the backup. This file increases in size for every backup operation. Remove the existing contents of the file if you do not require the details.

Restoring the Inventory Database

1. If the Inventory database server is up, stop the Storer service. At the database server console, enter **`StopSer Storer`**.
2. Exit the Sybase database.
   - On NetWare servers: At the database server prompt, enter **q** to stop the Sybase database.
   - On Windows 2000/2003: In the Windows Control Panel, double-click Administrative Tools, double-click Services, select Novell Database - Sybase, then click Stop.
3. Copy the backup files, overwriting the working database files.
4. Restart the database server.

Setting Up the Oracle Inventory Database

The following sections explain how to configure the Inventory database for Oracle8i and Oracle9i:

- "Creating the Oracle8i Inventory Database on a Windows Server" on page 471
- "Creating the Oracle9i Inventory Database on a Windows Server" on page 473
- "Creating the Oracle9i Inventory Database on a UNIX Server" on page 475
- "Manually Creating the Oracle Inventory Database Object" on page 476
- "Configuring and Running Multiple Oracle Database Instances on a Windows Server" on page 478

IMPORTANT: In this setup, the Inventory database is not mounted with any other version or instances of Oracle databases.

Creating the Oracle8i Inventory Database on a Windows Server

Ensure that the following prerequisites are met:

- ❑ Oracle 8.1.7 must be installed on the server.
- ❑ To maintain the Inventory database on Oracle, Server Inventory requires that you have a minimum of 25 user licenses.

You must manually create the Inventory database for Oracle on Windows servers by following the procedure below:

1. Create a directory c:\schema and copy all the files in the following directories from the *ZENworks 6.5 Companion 2 CD* to the schema directory:

   database\oracle8i\common

   database\oracle8i\winntspecific
2. By default _create.sql, init.ora, _start.sql are Read-only files. Make the files writable.
3. Create the *user_specified_path*\zenworks\inventory\oracle\database\trace directory structure.
4. In c:\schema_create.sql, replace all instances of **d**: with *user_specified_path*.

5 In c:\schema\init.ora, replace all instances of **d**: with *user_specified_path*.

6 In c:\schema_start.sql, replace all instances of **d**: with *user_specified_path*.

If **d**: is not found, check and correct the path of INIT.ORA in the database directory.

7 Copy c:\schema\init.ora to *user_specified_path*\zenworks\inventory\oracle\ database.

8 Copy c:\schema_start.sql to *user_specified_path*\zenworks.

9 Ensure that the Oracle services are loaded correctly and no database is mounted.

10 At the command prompt, enter **`svrmgrl`** to load the Oracle Server Manager.

11 At the Oracle Server Manager prompt (svrmgrl prompt), enter **`@c:\schema\schema.sql`**.

Review the c:\schema\inv.log file to ensure that the database has been created successfully. If the database has not been successfully created, inv.log will contain the following error messages: `Oracle not available, Out of space, Compilation error`.

12 Add non-English enumerated (enum) values for certain Inventory attributes into the Inventory database.

IMPORTANT: You must perform this step on the English version of the product also.

You need to add the non-English enumerated values so the Inventory report can display the enum value for the inventory attributes in internationalized versions. The non-English enum values must be available in English version of the product so that the rolled-up inventory information from non-English sites can be properly captured at the high-level servers where only English versions are installed.

For more information about the list of attributes that contains enumerated values, see Appendix K, “Enumeration Values,” on page 723.

To add the non-English enum values:

12a Specify the JDBC connection settings in the *zenworks_directory*\inv\server\wminv\properties\connection.prop file to connect to the Oracle database.

You can do this by copying the template property settings for Oracle specified in the comments section in the connection.prop file. Specify the IP address, port number, and Database SID in the JDBC URL string that matches your Oracle server configuration.

12b At the server prompt, enter **`AddEnums`** ***`directory_name_containing_connection.prop`***

If your Inventory server is running on a Windows machine, run the above command from *zenworks_directory*\inv\server\wminv\bin.

12c Execute the following SQL statement at the svrmgrl prompt to ensure that the localized enumerated values are added correctly:

- To display the enumerated values in French: **`connect mw_dba/`*`password`*** and **`SELECT * FROM cim.ostype_fr`**
- To display the enumerated values in Spanish: **`connect mw_dba/`*`password`*** and **`SELECT * FROM cim.ostype_es;`**
- To display the enumerated values in German: **`connect mw_dba/`*`password`*** and **`SELECT * FROM cim.ostype_de;`**
- To display the enumerated values in Brazilian-Portuguese: **`connect mw_dba/`*`password`*** and **`SELECT * FROM cim.ostype_pt_BR;`**

13 At the svrmgrl prompt, enter **`@c:\schema\schema1.sql`**.

Review the c:\schema\inv1.log file to ensure that the database has been created successfully. If the database has not been successfully created, inv1.log will contain the following error messages: `Oracle not available, Out of space, Compilation error`.

14 At the svrmgrl prompt, enter **`connect internal`** to login as DBA.

15 At the svrmgrl prompt, enter **`shutdown normal`**.

16 At the svrmgrl prompt, enter **`@`*`path`*`\zenworks\_start.sql`** to start the Inventory database.

17 Manually create the Inventory Database objects.

Creating the Oracle9i Inventory Database on a Windows Server

Ensure that the following prerequisites are met:

- Oracle9i release 2 must be installed on the server before configuring the Inventory database.
- To maintain the Inventory database on Oracle, Server Inventory requires that you have a minimum of 25 user licenses.

You must manually create the Inventory database for Oracle on Windows servers by following the procedure below:

1 Create a directory c:\schema and copy all the files in the following directories from the *ZENworks 6.5 Companion 2 CD* to the schema directory:

database\oracle9i\common
database\oracle9i\winntspecific

2 By default _create.sql, init.ora, _start.sql are Read-only files. Make the files writable.

3 Create the *user_specified_path*\zenworks\inventory\oracle\database\trace directory structure.

4 In c:\schema_create.sql, replace all instances of **d**: with *user_specified_path*.

5 In c:\schema\init.ora, replace all instances of **d**: with *user_specified_path*.

6 In c:\schema_start.sql, replace all instances of **d**: with *user_specified_path*.

If **d**: is not found, check and correct the path of init.ora in the database directory.

7 Copy c:\schema\init.ora to *user_specified_path*\zenworks\inventory\oracle\ database.

8 Copy c:\schema_start.sql to *user_specified_path*\zenworks.

9 Ensure that Oracle services are loaded correctly and the database is not mounted.

10 At the command prompt, enter **`sqlplus /nolog`** to load the Oracle Server Manager.

11 At the Oracle Server Manager prompt (sqlplus prompt), enter **`@c:\schema\schema.sql`**.

Review the c:\schema\inv.log file to ensure that the database has been created successfully. If the database has not been successfully created, inv.log will contain the following error messages: `Oracle not available, Out of space, Compilation error`.

12 Add non-English enumerated (enum) values for certain Inventory attributes into the Inventory database.

IMPORTANT: You must perform this step on the English version of the product also.

You need to add the non-English enumerated values so the Inventory report can display the enum value for the inventory attributes in internationalized versions. The non-English enum values must be available in English version of the product so that the rolled-up inventory information from non-English sites can be properly captured at the high-level servers where only English versions are installed.

For more information about the list of attributes that contains enumerated values, see Appendix K, “Enumeration Values,” on page 723.

To add the non-English enum values:

12a Specify the JDBC connection settings in the *zenworks_directory*\inv\server\wminv\properties\connection.prop file to connect to the Oracle database.

You can do this by copying the template property settings for Oracle specified in the comments section in the connection.prop file. Specify the IP address, port number, and Database SID in the JDBC URL string that matches your Oracle server configuration.

12b At the server prompt, enter **AddEnums *directory_name_containing_connection.prop***

If your Inventory server is running on a Windows machine, run the above command from *zenworks_directory*\inv\server\wminv\bin.

12c Execute the following SQL statement at the sqlplus prompt to ensure that the localized enumerated values are added correctly:

- To display the enumerated values in French: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_fr**
- To display the enumerated values in Spanish: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_es;**
- To display the enumerated values in German: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_de;**
- To display the enumerated values in Brazilian-Portuguese: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_pt_BR;**

13 At the sqlplus prompt, enter **@c:\schema\schema1.sql**.

Review the c:\schema\inv1.log file to ensure that the database has been created successfully. If the database has not been successfully created, inv1.log will contain the following error messages: `Oracle not available, Out of space, Compilation error`.

14 At the sqlplus prompt, enter **connect / as sysdba** to login as DBA.

15 At the sqlplus prompt, enter **shutdown immediate**.

16 At the sqlplus prompt, enter **@*path*\zenworks_start.sql** to start the Inventory database.

17 Continue with “Manually Creating the Oracle Inventory Database Object” on page 476.

Creating the Oracle9i Inventory Database on a UNIX Server

Make sure that the following prerequisites are met:

- Oracle 9.2.0.6 must be installed must be installed on Linux or Solaris versions supported by Oracle9i.
- Hard disk free space: 4 GB or above.
- Primary memory: 1 GB or above.
- To maintain the Inventory database on Oracle, Server Inventory requires that you have a minimum of 25 user licenses.

You must manually create the Inventory database for Oracle on UNIX servers by following the procedure below:

1. Log in as an Oracle user.
2. Create a /schema directory in the Oracle installation directory (by default, Oracle is installed in the /opt/oracle directory), and copy all the files in the following directories from the *ZENworks 6.5 SP2 Companion CD* to the schema directory:

   database\oracle9i_unix\unixspecific
   database\oracle9i_unix\common
3. By default, _create.sql, init.ora, _start.sql are Read-only files. Make the files writable.
4. Create the *user_specified_directory_path*/zenworks/inventory/oracle/database/trace directory structure in /opt/oracle.
5. In schema/init.ora, replace all instances of $HOME with *user_specified_directory_path*.
6. In schema/_start.sql, replace all instances of $HOME with *user_specified_directory_path*.
7. In schema/_create.sql, replace all instances of $HOME with *user_specified_directory_path*.
8. In schema/schema.sql, replace all instances of $HOME with the schema directory created in Step 2.
9. In schema/schema1.sql, replace all instances of $HOME with the schema directory created in Step 2.
10. Copy schema/init.ora to *user_specified_directory_path*/zenworks/inventory/oracle/database.
11. Copy schema/_start.sql to *user_specified_directory_path*.
12. Make sure the Oracle services are up and running and no database is mounted.
13. At the command prompt, enter **`sqlplus /nolog`** to load the Oracle Server Manager.
14. At the Oracle Server Manager prompt, enter **`@$HOME/schema/schema.sql`**, where $HOME is the schema directory created in Step 2.
15. Review the schema/inv.log file to make sure that the database has been created successfully. If the database has not been successfully created, inv.log will contain the following error messages: `Oracle not available, Out of space, Compilation error`.
16. Add non-English enumerated (enum) values for certain Inventory attributes into the Inventory database.

    IMPORTANT: You must perform this step on the English version of the product also.

    You need to add the non-English enumerated values so the Inventory ConsoleOne utilities such as Inventory Report can display the enum value for the inventory attributes in internationalized versions. The non-English enum values must be available in English version

of the product so that the rolled-up inventory information from non-English sites can be properly captured at the high-level servers where only English versions are installed.

To add the non-English enum values:

16a On the Inventory Server machine, Specify the JDBC connection settings in the *zenworks_directory*\inv\server\wminv\properties\connection.prop file to connect to the Oracle database.

You can do this by copying the template property settings for Oracle specified in the comments section in the connection.prop file. Specify the IP address, port number, and Database SID in the JDBC URL string that matches your Oracle server configuration.

16b At the server prompt, enter **AddEnums *directory_name_containing_connection.prop***.

If your Inventory server is running on a Windows machine, run the above command from *zenworks_directory*\inv\server\wminv\bin.

16c Execute the following SQL statement at the sqlplus prompt to make sure that the localized enumerated values are added correctly:

- To display the enumerated values in French: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_fr**
- To display the enumerated values in Spanish: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_es;**
- To display the enumerated values in German: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_de;**
- To display the enumerated values in Brazilian-Portuguese: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_pt_BR;**
- To display the enumerated values in Japanese: **connect mw_dba/*password*** and **SELECT * FROM cim.ostype_ja;**

17 At the sqlplus prompt, enter **@$HOME/schema/schema1.sql**, where $HOME is schema directory created in Step 2.

Review the schema/inv1.log file to make sure that the database has been created successfully. If the database has not been successfully created, inv1.log contains the following error messages: `Oracle not available, Out of space, Compilation error`.

18 At the sqlplus prompt, enter **connect / as sysdba** to login as DBA.

19 At the sqlplus prompt, enter **shutdown immediate**.

20 At the Oracle Server Manager prompt, enter **@*user_specified_directory_path*/zenworks/_start.sql** to start the Inventory database.

Manually Creating the Oracle Inventory Database Object

1 In ConsoleOne, right-click a location in the Novell eDirectory™ tree for the database object, then click New, click Object, click ZENworks Database, then click OK.

2 Type a name for the database object, then click OK.

3 Configure the database server options of the database object.

3a In ConsoleOne, right-click the database object (Inventory database_*server_name*), then click Properties, then click the ZENworks Database tab.

3b Select the database server object by using either of the following methods:

- If eDirectory is installed on the database server, then in the Server DN field, browse for and select the Server object of the server where the database is physically installed and running.

  The server's IP address is automatically populated to the Server IP Address or DNS Name drop-down list. If the selected server object has more than one IP address, select the appropriate IP address.

  IMPORTANT: Ensure that the DNS name of the database server configured for the database object is valid. If the DNS name is invalid, you must select an appropriate database server IP address in the Database object property page.

  To clear the value set in the Server DN field, type the IP address of another database server or browse and select another server object.

- If eDirectory is not installed on the database server, then specify the server IP address or the DNS name in the Server IP Address or DNS Name field.

3c Specify the following values:

- **Database (Read-Write) User Name:** *MW_DBA*
- **Database (Read-Write) Password:** *novell*
- **Database (Read Only) User Name:** *MWO_READER*
- **Database (Read Only) Password:** *novell*
- **Database (Write Only) User Name:** *MWO_UPDATER*
- **Database (Write Only) Password:** *novell*

IMPORTANT: All Inventory components use the username and the password configured in the database object. By default, "novell" is the password for all options. But you can change it in the database, and update the same here.

3d Click Apply.

3e To configure the JDBC Driver properties, click the JDBC Driver Information tab.

3f Select Oracle, then click the Default Settings button.

This populates the fields with default JDBC driver information.

The database settings for Oracle are:

- **Driver:** *oracle.jdbc.driver.OracleDriver*
- **Protocol:** *jdbc:*
- **SubProtocol:** *oracle*:
- **SubName:** *thin:@*
- **Port:** *1521*
- **Flags:** This field is not applicable for Oracle
- **Database Service Name:** *orcl* (The value for the SID is the same as assigned for the database instance.)

3g Click Apply, then click Close.

Configuring and Running Multiple Oracle Database Instances on a Windows Server

Ensure that the following prerequisites are met:

- ❑ Oracle9i release 2 must be installed on the Windows Inventory server.
- ❑ To maintain the Inventory database in Oracle, Server Inventory requires that you have a minimum of 25 user licenses.
- ❑ You have already set up the Inventory database.

To configure and run Oracle instances:

1. At the database server, from the desktop Start menu, click Programs, click Oracle, click Database Administration, then click Oracle Database Configuration Assistant.
2. Click Create a Database, click Next, click Typical, click Next, click Copy Existing Database Files from the CD, then click Next.
3. Enter the following details:
   - **Global Database Alias:** `mgmtdb.`*your_windows_nt/2000_name*
   - **SID:** By default, the value is *mgmtdb*.
4. Click Finish.

   This process takes a significant amount of time and creates the Oracle database.

   Ensure that the OracleServiceMGMTDB service is created and started.
5. Load the Inventory database.

   From the desktop menu, click Start, click Run, then click SQLPLUS to run the Oracle Server Manager.

   Enter the following commands:

   ```
   set instance mgmtdb
   connect internal/password_for_administrator
   ```

Setting Up the MS SQL Server 2000 Inventory Database

This section provides information on the following topics:

- “Configuring the MS SQL Server 2000 Inventory Database” on page 479
- “Manually Creating the MS SQL Server 2000 Inventory Database Object” on page 481
- “Connecting the Inventory Server and ConsoleOne to the MS SQL Server 2000 Inventory Database” on page 482

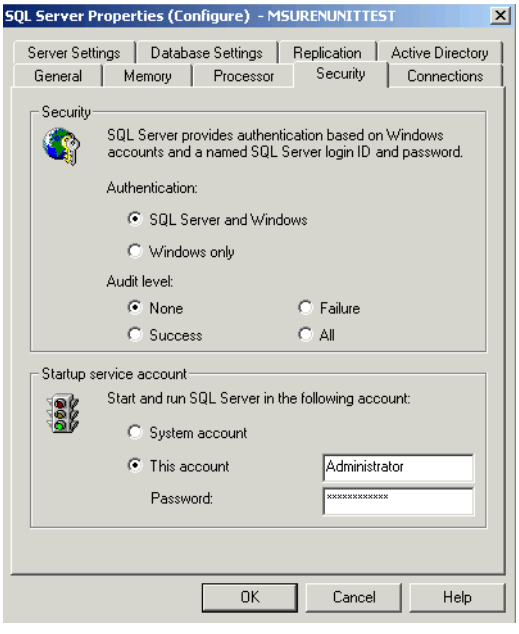
Configuring the MS SQL Server 2000 Inventory Database

Prerequisites for configuring the database include the following:

- Microsoft* SQL Server 2000 installed on the Windows server.
- Minimum free disk space of 50 MB to extract the p1mssqlinvdb.zip file.
- Ensure that you have sufficient disk space to store the inventory information on the server that has the Inventory database.

To configure the Inventory database for MS SQL Server 2000:

1. Copy the p1mssqlinvdb.zip file from the *ZENworks 6.5 Companion 2* CD\database\mssql directory to *path_of_inventory_database_directory_on_the_database_server*.
2. Extract p1mssqlinvdb.zip.
3. From the MS SQL server desktop Start menu, click Programs, click Microsoft SQL Server, then click Enterprise Manager.
4. In the SQL Server Enterprise Manager, browse to Console Root/Microsoft SQL Servers/SQL Server Group/*machine_name_running_Inventory_database*.
5. Right -click *machine_name_running_Inventory_database*, then click Properties.
6. In the SQL Server Properties dialog box, click the Security tab and ensure that the authentication is set to SQL Server and Windows.

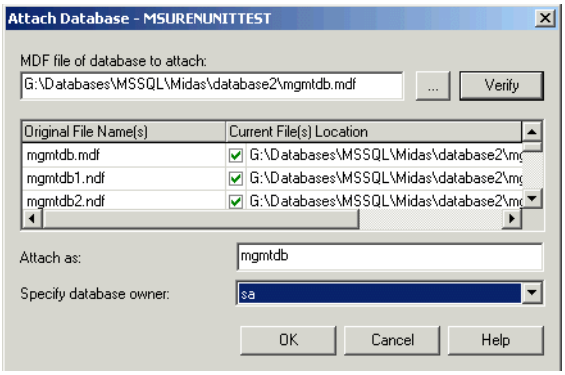


7. Click OK.
8. Browse to *machine_name_running_Inventory_database*/Databases and right-click Databases, click All Tasks, then double-click Attach Database.
9. In the Attach Database dialog box, do the following:
   - 9a Click the Browse button to browse to and select mgmtdb.mdf as the .mdf database file to be attached.
   - 9b Ensure that the value of the Attach As field is mgmtdb.

9c Select sa from the Specify database owner drop-down list.

9d Click OK.

The ZENworks Inventory database (mgmtdb) is attached to the Databases server group.



10 Select mgmtdb, then click the Tools menu, then click SQL Query Analyzer.

11 In the SQL Query Analyzer, do the following:

11a Ensure that mgmtdb is selected in the drop-down list.

11b Click File, then click Open.

11c Select the createloginnames.sql query file from the *ZENworks 6.5 Companion 2* CD\database\mssql directory.

11d Click Query, then click Execute.

On successful execution, the following message is displayed in the Message pane:

```
New Login Created
```

11e Login as MW_DBA in the SQL Query Analyzer and execute the following drop trigger sqls:

```
drop trigger cim.x$cim$component

go

drop trigger cim.x$cim$dependency

go

drop trigger managewise.x$managewise$designates

go

drop trigger managewise.x$managewise$currentloginuser

go

drop trigger managewise.x$managewise$lastloginuser

go

drop trigger cim.x$cim$installedsoftwareelement

go
```

NOTE: During the execution of the drop trigger sqls, the following error message might be displayed on the console, “Cannot drop the trigger '*trigger_name*', because it does not exist in the system catalog”. Ignore the error message.

12 (Optional) Add non-English enumerated (enum) values for certain Inventory attributes into the Inventory database.

You need to add the non-English enumerated values so the Inventory report can display the enum value for the inventory attributes in internationalized versions. The non-English enum values must be available in English version of the product so that the rolled-up inventory information from non-English sites can be properly captured at the high-level servers where only English versions are installed.

For more information about the list of attributes that contains enumerated values, see Appendix K, “Enumeration Values,” on page 723.

To add the non-English enum values:

12a Specify the JDBC connection settings in the *zenworks_directory*\inv\server\wminv\properties\connection.prop file to connect to the MS SQL database.

You can do this by copying the template property settings for MS SQL specified in the comments section in the connection.prop file. Specify the IP address, port number, and Database SID in the JDBC URL string that matches your MS SQL server configuration.

12b At the server prompt, enter **AddEnums *directory_name_containing_connection.prop***

If your Inventory server is running on a Windows machine, run the above command from *zenworks_directory*\inv\server\wminv\bin.

After executing the command, the a message indicating that the non-English enums have been successfully inserted is displayed on the console prompt.

13 Continue with “Manually Creating the MS SQL Server 2000 Inventory Database Object” on page 481.

Manually Creating the MS SQL Server 2000 Inventory Database Object

1 In ConsoleOne, right-click a location in the Novell eDirectory tree for the database object, then click New, click Object, click ZENworks Database, then click OK.

2 Type a name for the database object, then click OK.

3 Configure the database server options of the database object.

3a In ConsoleOne, right-click the database object (Inventory database_*server_name*), then click Properties, then click the ZENworks Database tab.

3b Select the database server object by using either of the following methods:

- If eDirectory is installed on the database server, then in the Server DN field, browse for and select the Server object of the server where the database is physically installed and running.

  The server’s IP address is automatically populated to the Server IP Address or DNS Name drop-down list. If the selected server object has more than one IP address, select the appropriate IP address.

  IMPORTANT: Ensure that the DNS name of the database server configured for the database object is valid. If the DNS name is invalid, you must select an appropriate database server IP address in the Database object property page.

To clear the value set in the Server DN field, type the IP address of another database server or browse and select another server object.

- If eDirectory is not installed on the database server, then specify the server IP address or the DNS name in the Server IP Address or DNS Name field.

3c Specify the following values:

- **Database (Read-Write) User Name:** *MW_DBA*
- **Database (Read-Write) Password:** *novell*
- **Database (Read Only) User Name:** *MWM_READER*
- **Database (Read Only) Password:** *novell*
- **Database (Write Only) User Name:** *MWM_UPDATER*
- **Database (Write Only) Password:** *novell*

IMPORTANT: All Inventory components use the username and the password configured in the database object. By default, “novell” is the password for all options. But you can change it in the database, and update the same here.

3d Click Apply.

3e To configure the JDBC Driver properties, click the JDBC Driver Information tab.

3f Select MSSQL, then click the Default Settings button.

This populates the fields with default JDBC driver information.

The database settings for MS SQL are:

- **Driver:** *com.microsoft.jdbc.sqlserver.SQLServerDriver*
- **Protocol:** *jdbc:*
- **SubProtocol:** *microsoft*:
- **SubName:** *sqlserver://*
- **Port:** *1433*
- **Flags:** This field is not applicable for MS SQL.
- **Database Service Name:** This field is not applicable for MS SQL.

3g Click Apply, then click Close.

4 Continue with “Connecting the Inventory Server and ConsoleOne to the MS SQL Server 2000 Inventory Database” on page 482.

Connecting the Inventory Server and ConsoleOne to the MS SQL Server 2000 Inventory Database

The Inventory server components and ConsoleOne use the Microsoft JDBC driver to connect to the Inventory database on MS SQL 2000. You must install and configure the Microsoft SQL Server 2000 driver for JDBC to use the Inventory system.

To configure the Microsoft SQL Server 2000 driver for JDBC to access the Inventory database running on MS SQL 2000:

1 Download the Windows English version of Microsoft JDBC driver from the Microsoft SQL Server web site (http://www.microsoft.com/downloads/details.aspx?FamilyID=9f1874b6-f8e1-4bd6-947c-0fc5bf05bf71&DisplayLang=en).

2 Install the driver on a Windows machine.

3 Copy the msbase.jar, msutil.jar, and mssqlserver.jar files to the *inventory_server_installation_directory*\inv\server\lib directory.

4 On the machine running ZENworks 6.5 Server Management ConsoleOne with Inventory snap-ins, copy the msbase.jar, msutil.jar, and mssqlserver.jar files to the *consoleone_installation_directory*\lib\zen directory.

5 In ConsoleOne, create a database object in the same container where the Inventory server is installed.

5a Right-click the container.

5b Click New, click Object, select ZENworks Database from the list of objects, then click OK.

5c Enter a name for the database object, then click OK.

6 Configure the Database server options of the Database object.

6a In ConsoleOne, right-click the database object, click Properties, then click the ZENworks Database tab.

6b Select the database server object using any of the following methods:

- If eDirectory is installed on the database server, in the Server DN field, browse for and select the Server object for the server where the database is physically installed and running.

  The server's IP address is automatically populated to the Server IP Address or DNS Name drop-down list. If the selected server object has more than one IP address, select the appropriate IP address.

  IMPORTANT: Ensure that the DNS name of the database server configured for the database object is valid. If the DNS name is invalid, you must select an appropriate database server IP address in the Database object property page.

  To clear the value set in the Server DN field, type the IP address of another database server or browse and select another server object.

- If eDirectory is not installed on the database server, specify the server's IP address or the DNS name in the Server IP Address or DNS Name field.

6c Type the values for the following options:

- **Database (Read-Write) User Name:** *MW_DBA*
- **Database (Read-Write) Password:** *novell*
- **Database (Read Only) User Name:** *MWM_READER*
- **Database (Read Only) Password:** *novell*
- **Database (Write Only) User Name:** *MWM_UPDATER*
- **Database (Write Only) Password:** *novell*

6d Click Apply.

6e To configure the JDBC Driver properties, click the JDBC Driver Information tab.

6f Select MS SQL, then click Default Settings.

This populates the fields with default JDBC driver information.

Modify the database settings based on the configuration of your MS SQL Server. The database settings for MS SQL are:

- **Driver:** *com.microsoft.jdbc.sqlserver.SQLServerDriver*
- **Protocol:** *jdbc:*
- **SubProtocol:** *microsoft*:
- **SubName:** *sqlserver://*
- **Port:** *1433*
- **Flags:** Not applicable for MS SQL
- **Database Service Name:** Not applicable for MS SQL

6g Click Apply, then click Close.

Configuring the Inventory Service Object

The Inventory Service object settings configure the scanning for the associated inventoried servers.

To configure the Inventory Service Object:

1. In ConsoleOne, right-click the Inventory Service object (Inventory Service_*server_name*), then click Properties to display the Inventory Service Object Properties page.
2. Modify the following settings:

   Inventory Server Role: Based on the Inventory servers that you have deployed for scanning inventory, you must specify the role of the Inventory server. See "Understanding the Inventory Server Roles" on page 420.

   Discard Scan Data Time: Any scan data files that have scan information collected before the Discard Scan Data Time that you specify in the Inventory Service Object Property page will be discarded. The scan data files are removed from the Inventory server, which is one of the following types: Intermediate Server, Intermediate Server with Database, Intermediate Server with Database and Inventoried Servers, and Intermediate Server with Inventoried Servers.

   Scan Directory Path: The directory on the Inventory server where scans received from the inventoried server or rolled up from other Inventory servers will be stored for further processing. On a NetWare Inventory server, the scan directory path is *complete_DNS_name_of_the_server**directory_path_of_scandir*. For example, *complete_DNS_name_of_the_server*\sys:\zenworks\inv\scandir. On a Windows Inventory server, the scan directory path is *complete_DNS_name_of_the_server*\scandir.

   You cannot modify the Inventory Server name specified in the \scandir path. But if you want modify the directory, make sure that the new directory already exists before changing the scan directory path.

   To modify the path:

   - On NetWare, click Browse to select and add the path or enter the path.
   - On Windows, you must manually enter the path.

   Enable Scan: To scan the inventoried servers associated with the Inventory Service object, you must enable the scan option listed in the Inventory Service Object property page. To disable the scanning of inventoried servers, deselect this option.

3 Click Dictionary Settings tab and configure the required software dictionary rules. For more information, see “Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 580.

4 Click OK.

NOTE: If you are modifying the Inventory policies or configuring the objects, always stop the Inventory services. Configure the policies and properties of the objects. Restart the Inventory services again. For more information, see “Starting and Stopping the Inventory Service” on page 454.

Configuring the Database Location Policy

The Database Location policy contains the location of the Inventory database. You can associate the Database object with a container under which the Inventory Service object is located through using the Service Location Package or with an Inventory server through using the Server Package.

NOTE: If you configure the Service Location Package and the Server Package, the Server Package settings will override the Service Location Package settings.

To associate the Database object with a container under which the Inventory Service object is located:

1 In ConsoleOne, right-click the Service Location Package, then click Properties to display the Policies page.

2 Select the check box under the Enabled column for the ZENworks Database policy.

3 Click Properties to display the Inventory Management page.

4 Browse to the DN of the Inventory Database object (Inventory database_*server_name*), then click OK.

For a Sybase database, the database object is automatically created during the Server Inventory installation unless you are installing on a Windows server without eDirectory installed. To manually create the database object, see “Manually Creating the Sybase Inventory Database Object” on page 466.

For an Oracle database, you must create the database object and configure the object. For more information, see “Setting Up the Oracle Inventory Database” on page 471.

For an MS SQL database, you must configure the database object. For more information, see “Setting Up the MS SQL Server 2000 Inventory Database” on page 478.

5 Click OK.

6 Click the Associations tab, then click Add.

7 Browse to select the container under which the Inventory Service object is located, then click OK.

8 Click Apply, then click Close.

To associate the Database object with an Inventory server:

1 In ConsoleOne, right-click the Server Package, click Properties to display the Policies page.

2 Select the check box under the Enabled column for the ZENworks Database policy.

3 Click Properties to display the Inventory Management page.

4 Browse to the DN of the Inventory Database object (Inventory database_*server_name*), then click OK.

For a Sybase database, the database object is automatically created during the Server Inventory installation unless you are installing on a Windows server without eDirectory installed. To manually create the database object, see "Manually Creating the Sybase Inventory Database Object" on page 466.

For an Oracle database, you must create the database object and configure the object. For more information, see "Setting Up the Oracle Inventory Database" on page 471.

For an Oracle database, you must create the database object and configure the object. For more information, see "Setting Up the Oracle Inventory Database" on page 471.

5 Click OK.

6 Click the Associations tab, then click Add.

7 Browse to select an Inventory server object, then click OK.

8 Click Apply, then click Close.

NOTE: If you are modifying the Inventory policies or configuring the objects, always stop the Inventory services. Configure the policies and properties of the objects. Restart the Inventory services again. For more information, see "Starting and Stopping the Inventory Service" on page 454.

Configuring the Server Inventory Policy

The Server Inventory policy contains the IP address or the DNS name of the Inventory server to which the inventory information is sent. This policy also contains the inventory scanning schedule for the associated inventoried server. You must configure the Server Inventory policy for each inventoried server.

To configure the Server Inventory policy package:

1 In ConsoleOne, right-click the Distributed Server Package, then click Properties to display the Policies page.

2 Click the Policies tab, then click NetWare or Windows from the drop-down list, depending on the operating system of the inventoried server.

3 Select the check box under the Enabled column for the Server Inventory policy.

4 Click Properties to display the Server Inventory Policy page.

5 In the General tab, configure the following settings:

5a Browse to select the DN of the Inventory Service object (Inventory Service_*server_name*).

This setting specifies that the scanner will send the server inventory information to this Inventory server.

5b Select the DNS name or the IP address of the Inventory server.

5c If the scan is to an Inventory server that is across the firewall, specify the IP address and the port number of the proxy server.

6 (Optional) Customize the Inventory scan.

6a To customize the hardware scan for the Windows inventoried servers, click the Hardware Scan tab and configure the following settings:

Enable DMI Scan: Includes DMI scanning of Windows inventoried servers.

Enable WMI Scan: Includes WMI scanning of Windows inventoried servers.

6b To customize the software scan for the NetWare or Windows inventoried servers on which Novell ZENworks for Servers 3.0 or ZENworks for Servers 3.0.2 is installed, click the Software Scan tab and configure the following settings. For more information, see "Customizing the Software Inventory Information To Be Scanned For ZENworks for Servers 3.X Inventoried Servers" on page 615.

IMPORTANT: Do not configure these settings for the inventoried servers that have ZENworks 6.5 Server Management installed. To configure the software scan settings for ZENworks 6.5 Desktop Management inventoried workstations, see "Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers" on page 550.

Enable Software Scan: Enables software scanning for the inventoried servers associated with the Inventory policy. The scan program collects software information for the inventoried servers and stores it in the Inventory database.

Custom Scan Editor: Enables you to customize the list of application details to scan for at the Windows inventoried servers. The Inventory scanner scans for the details of the applications listed in the Custom Scan Editor.

For example, specify the following details in the Custom Scan Editor: Vendor Name=Microsoft; Product Name=Microsoft Office; Product Version=10.0; FileName=winword.exe; File Size=1 MB. The Inventory scanner scans for the winword.exe file having a size of 1 MB on the inventoried servers. If the file is found, the scanner stores "Microsoft;Microsoft Office;10.0" for "winword.exe;1 MB" in the Inventory database.

Product Identification Number: Enables you to scan for the product identification number of the Microsoft applications installed on the Windows inventoried servers only

6c Click the Configuration Editor tab; if required, modify the settings of the following .ini files.

- **Asset Information:** Scans for vendor-specific information from DMI. For more information on how to configure the Asset Information, see "Scanning for Vendor-Specific Asset Information from DMI" on page 546.
- **Zipped Names:** Customizes the hardware scanning of Jaz* and Zip* drives. For more information, see "Customizing the Hardware Scanning Information of Jaz and Zip Drive Vendors" on page 548.
- **SWRules:** Configure the SWRules file for the Windows inventoried servers on which Novell ZENworks for Servers 3.0 or ZENworks for Servers 3.0.2 is installed. Do not configure the settings for inventoried servers that have ZENworks 6.5 Server Management installed.

  The SWRules option customizes the software scanning information of vendors and products. For more information, see "Customizing the Software Inventory Information To Be Scanned For ZENworks for Servers 3.X Inventoried Servers" on page 615.
- **HWRules:** Customizes the nominal size of monitors. For more information on how to configure the HWRules .ini file, see "Customizing the Hardware Information for Monitor's Size" on page 549.

7 Click the Policy Schedule tab.

8 Modify the schedule, click Apply, then click Close.

9 In the Distributed Server Package property page, click the Distribution tab, then click Add.

10 Browse to add the Distribution object, then click OK.

11 Click Apply, then click Close.

12 In ConsoleOne, right-click the Inventory Service object, click Properties, then click the Inventory Service Object Properties tab.

13 Ensure that the Enable Scan of Machines check box is selected, then click OK.

This setting ensures that scanning is selected for the inventoried servers associated with the selected Inventory server.

Configuring the Roll-Up Policy

The Roll-Up policy settings configure the selected Inventory server for roll-up of scan information. The settings in the Roll-Up policy identify the next-level Inventory server (DN of the Inventory Service object) for moving the inventory information from the selected Inventory server. These settings stored in eDirectory are associated with the Inventory Server object.

To configure the Roll-Up policy:

1 In ConsoleOne, right-click the Server Package, click Properties, click Policies, then select the appropriate sub-option. If you want to this policy to be applied on all servers, select the General sub-option.

2 Check the check box under the Enabled column for the Inventory Rollup Policy.

3 Click Properties to display the Roll-Up Policy page.

4 Browse to select the DN of the Inventory Service object (Inventory Service_*server_name*).

Destination Server Object: You must specify the DN of the Inventory Service object at the next level Inventory server for moving the inventory information from the selected Inventory server. The server that you specify must be another Intermediate Server, Intermediate Server with Database, Intermediate Server with Database and Inventoried Servers, Intermediate Server with Inventoried Servers, Root Server, or Root Server with Inventoried Servers.

NOTE: Ensure that the specified Inventory server is a different server because you cannot roll-up of information to the same Inventory server. Also, you cannot specify the lower-level Inventory server as the next-destination server for roll-up of information.

5 By default, the DNS name or the IP address (if a DNS name is not configured) of the next-level server is populated in the field. If the next-level server has multiple IP addresses, select the preferred address.

IMPORTANT: Ensure that the DNS name of the next-level server is valid. If the DNS name is invalid, you must select an appropriate server IP address.

6 If the roll-up is to an Inventory server that is across the firewall, specify the IP address or the DNS name and the port number of the proxy server.

7 Click Apply.

8 Click the Roll-Up Policy tab, then click Roll-Up Schedule.

9 Modify the settings for scheduling the roll-up time and click Apply.

When you schedule the roll-up of information in the Inventory policies, we recommend the roll-up frequency should be at least one day. It is likely that if the roll-up of inventory information is scheduled too frequently, for example less than one hour, there might be some performance degradation of the Inventory server.

10 (Conditional) If you have not yet associated the Server Package, you will be prompted to associate it with an Inventory server or a container. The policy you configured and enabled earlier will not be in effect until you associate this policy package with an Inventory server or a container.

To associate the policy package:

10a Click the Associations tab, then click Add.

10b Browse for and select the Inventory server or the container that you want to associate the Roll-Up policy to.

10c Click OK, then click OK.

11 Click Apply, then click Close.

NOTE: If you are modifying the Inventory policies or configuring the objects except for the Roll-Up schedule, always stop the Inventory services. Configure the policies and properties of the objects. Restart the Inventory services again. For more information, see "Starting and Stopping the Inventory Service" on page 454.

Configuring the Dictionary Update Policy

The Dictionary Update policy configures the Inventory server to receive the software dictionary updates from other Inventory servers. You must manually download the dictionary updates to at least one Inventory server in your network. This Inventory server can then act as the source of dictionary updates to other Inventory servers.

1 In ConsoleOne, right-click the Server Package, click Properties, click Policies, then select the appropriate sub-option. If you want to this policy to be applied on all servers, select the General sub-option.

2 Select the check box under the Enabled column for the Dictionary Update policy.

3 Click Properties to display the Dictionary Update Policy page.

4 Configure the following settings:

4a (Recommended) Select the "Use the Roll-Up Server as the Update Source" check box if you want the Dictionary Update Service to use the Inventory server configured in the Roll-Up policy as the source for dictionary updates. If you do not select this option, the Dictionary Update Service will use the following settings.

If you select this check box, continue with Step 9 on page 490. If you do not select this option, the Dictionary Update Service will use the following settings configured in this policy (Dictionary Update Policy); continue with Step 4b on page 489.

NOTE: Do not select this option for a Standalone Server and a Root Server. You must manually configure the following settings of the policy.

4b In the Source Service Object field, browse to select the DN of the Inventory server, which provides the dictionary updates.

4c Select the IP address or the DNS name of the Inventory server, which provides the dictionary updates.

4d If the source Inventory server is across the firewall, specify the IP address or the DNS name and the port number of the XML proxy server.

4e Click Apply.

5 Click the Dictionary Update Policy tab, then click Dictionary Update Schedule.

6 Configure the Dictionary Update Schedule page to establish the schedule for running the Dictionary Consumer.

   We recommend you to configure the Weekly schedule.

7 Click Apply.

8 (Conditional) If you have not yet associated the Server Package, you will be prompted to associate it with an Inventory server or a container. The policy you configured and enabled earlier will not be in effect until you associate this policy package with an Inventory server or a container.

   To associate the policy package:

   8a Click the Associations tab, then click Add.

   8b Browse for and select the Inventory server or the container that you want to associate the Dictionary Update policy to.

   8c Click OK, then click OK.

9 Click Apply, then click Close.

NOTE: If you want to modify the Dictionary Update policy settings, you need not stop the Inventory services.

Setting Up Distribution of Dictionary

A software dictionary can be updated in the following ways:

- Manually download the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) to each Inventory server.

  NOTE: The dictionary is updated and published once every three months in this TID.

- Manually download the dictionary from the Novell Support web site to one Inventory server (preferably, the Root Server) and automatically distribute the dictionary to all servers in your setup by configuring the Dictionary Update policy.

An Inventory server can receive dictionary updates from any other Inventory server, irrespective of the server's role. The role of the Inventory server indicates whether the server receives the inventory information, stores the information into a local Inventory database, or rolls up the inventory information.

To update and distribute the software dictionary between Inventory servers:

1 Manually download the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) and save it in the *zenworks_installation_directory*\zenworks\inv\server\dictdir directory on the Inventory server.

2 Configure the Dictionary Update policy. For more information see, "Configuring the Dictionary Update Policy" on page 489.

All Inventory servers have Dictionary Provider and Dictionary Consumer services that are automatically installed during the Server Inventory installation.

When an Inventory server is started, the Dictionary Consumer reads the Dictionary Update policy and contacts the Dictionary Provider (running on another Inventory server) specified in the policy.

Subsequently, the Dictionary Consumer checks for the dictionary updates based on the schedule set in the Dictionary Update policy. It compares the date of the dictionary file on the Dictionary Provider with the file that has been locally stored. If the file on the Dictionary Provider is latest,

then the Dictionary Consumer downloads the file from the Dictionary Provider using XML-RPC as per the schedule.

The user-defined rules in the downloaded dictionary file are merged with the rules present in the local dictionary. If the merge yields a different set of rules from those locally present, the consolidated set of rules is written to the local dictionary. During the merge process, conflicts might arise, which are resolved on the basis of the following considerations:

- The rules in the downloaded dictionary always override the rules in the local dictionary.
- If a conflict arises between the software identifiers, the conflicting identifiers in the local dictionary are removed from the final (merged) dictionary.
- For a software dictionary rule, the final result is obtained by first writing the downloaded rules and then the local rules into the final dictionary; eliminating the duplicates during the process. This ensures that the downloaded software rules precede the local rules.

The following scenario illustrates the distribution of the software dictionary among the Inventory servers. In this scenario, there is an Inventory tree consisting of one Root Server (R1), one Leaf Server (L1), and two Standalone servers (S1 and S2). L1 rolls up the inventory information to R1.

Follow the below procedure to update the software dictionary on all the Inventory servers.

1. Manually download the dictionary on R1 from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp).
2. Create and configure a Dictionary Update policy by specifying R1 as the Dictionary Provider, and associate the policy to L1. For more information on how to configure the Dictionary Update policy, see “Configuring the Dictionary Update Policy” on page 489.
3. For S1 and S2, you can either manually download the dictionary from TID 10093255 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp) or configure the Inventory servers to automatically receive the latest version of the dictionary from R1.

   For S1 and S2 to automatically receive the latest version of the dictionary from R1, create and configure a Dictionary Update policy by specifying R1 as the Dictionary Provider, and associate the policy to S1 and S2. For more information on how to configure the Dictionary Update policy, see “Configuring the Dictionary Update Policy” on page 489.

14 Understanding the Server Inventory Components

The following sections describe the Novell® ZENworks® 6.5 Server Inventory components and processes:

Understanding the Inventory Service Manager

The Inventory Service Manager loads the inventory components on the Inventory server, based on the configuration parameters specified in the Inventory server properties file.

This section contains the following:

List of Services

The Service Manager loads the following important services. You can obtain the list of services that the Service Manager loads from the property file in *inventory_server_installation_directory_or_volume*\zenworks\inv\server\wminv\properties.

Server Configuration Service

Inventory Scheduler Service

Inventory Scheduler Service

Selector Service

Receiver Service

Sender Service
Storer Service
Scan Collector Service
Dictionary Consumer Service
Dictionary Provider Service

You can use these service names to list, start, and stop the respective services.

The Inventory Service Manager reads the server property file (config.properties) and the role-based property file in the *inventory_server_installation_directory_or_volume*\zenworks\inv\server\wminv\properties directory, and loads the required services and server components.

IMPORTANT: Do not modify the property files because the updates might fail to load the services or the Service Manager.

Services on NetWare Inventory Servers

On a NetWare® Inventory server, the installation program modifies the autoexec.ncf file located in sys:\system directory to load startinv.ncf. The startinv.ncf file located in the sys:\system brings up the Inventory Service Manager at Inventory server startup time.

You can start, stop, or list the services, if the Inventory Service Manager is already loaded.

- To check if the Inventory Service Manager is loaded, at the server prompt, enter **`java -show`**.

  This will display the following message:

  ```
  com.novell.zenworks.inventory.servercommon.ZENWorksInventoryServiceManager
  ```

- To start an Inventory service, enter **`StartSer service_name`** at the Inventory server prompt.

  service_name refers to any of the listed services. Follow the service naming syntax when you modify the *service_name*.

  For example, to start the Storer, enter **`StartSer Storer`**

- To stop an Inventory service, enter **`StopSer service_name`** at the Inventory server prompt.

  service_name refers to any of the listed services. Follow the service naming syntax when you modify the s*ervice_name*.

  For example, to stop the Storer, enter **`StopSer Storer`**

- To list an Inventory service, enter **`ListSer service_name`** at the Inventory server prompt.

  service_name refers to any of the listed services. Follow the service naming syntax when you modify the s*ervice_name*.

- To list all Inventory services, enter **`ListSer *`** at the Inventory server prompt.

Services on Windows Inventory Servers

On Windows* Inventory servers, the installation program creates the Service Manager as a service. During server startup, this Inventory Service Manager is loaded as a service.

You can start, stop, or list the services, if the Inventory Service Manager (ZENworks Inventory Service) is already loaded.

To start the Inventory service on the Windows 2000/2003 Inventory server:

1. In the Control Panel, double-click Administrative Tools.
2. Double-click Services.
3. Select Novell Inventory Service, then click Start.

To stop the Inventory service on the Windows 2000/2003 Inventory server:

1. In the Control Panel, double-click Administrative Tools.
2. Double-click Services.
3. Select Novell Inventory Service, then click Stop.

To start an Inventory service:

1. Go to the *installation_directory*\inv\server\wminv\bin directory.
2. At the prompt, enter **`StartSer service_name`**.

   where *service_name* refers to an Inventory service.

To stop an Inventory service:

1. Go to the *installation_directory*\inv\server\wminv\bin directory.
2. At the prompt, enter **`StopSer service_name`**.

   where *service_name* refers to an Inventory service.

To stop all Inventory services (ZENworks Inventory Service), use the Windows services from the desktop menu.

To list an Inventory service:

1. Go to the *installation_directory*\inv\server\wminv\bin.
2. At the prompt, enter **`ListSer [-verbose] service_name`**.

   where *service_name* refers to an Inventory service.

Follow the service naming syntax when you modify the service_name.

To refer to all services, use the asterisk (*) wildcard character within double quotes “*”. This wildcard character can be used with ListSer parameters.

Understanding the Server Configuration Service

The Server Configuration Service performs the following tasks:

- Reads the policy information from the Novell eDirectory™ and passes it to other Inventory components.
- Validates the policies to ensure that the policies are correctly configured.
- Validates the Inventory database version.

Understanding the Inventory Scanner

ZENworks 6.5 Server Management uses the Inventory scanner to collect hardware and software information from NetWare and Windows inventoried servers.

The scanners collect hardware details such as: floppy disk drive, hard disk drive, BIOS, bus, mouse, keyboard, display adapters, network adapter cards, modems, Jaz* drives, Zip* drives, sound cards, memory cards, serial ports, parallel ports, processors, and modems. The software scanning includes checking for applications on the inventoried servers and reporting the information about the scanned software, such as the vendor name, the product name and version, etc.

The following sections contain detailed information about the Inventory scanners:

- “Inventory Scanning Process” on page 496
- “Scanning for the NetWare Inventoried Servers” on page 497
- “Scanning for the Windows Inventoried Servers” on page 499

You can customize the hardware information and the software information to be scanned. For more information, see “Customizing the Hardware Inventory Information To Be Scanned” on page 546 and “Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers” on page 550.

Inventory Scanning Process

1. Subscriber must be installed and configured on the inventoried servers.
2. The Server Inventory policy lets you configure the scanning schedule based on which the policy engine schedules and enforces scanning at the inventoried servers.
3. The Inventory scanner checks whether an updated dictionary is available at its Inventory server and downloads the updated dictionary.
4. The Policy Enforcer first reads the inventory settings configured in the Inventory Service object and the Server Inventory policy, and then launches the Inventory scanner.
5. The scanner scans for the hardware and software information.
6. The scan information collected by the scanners is stored as scan data files (.str). The files are sent to the Inventory server.

Scanning for the NetWare Inventoried Servers

The Inventory scanner scans for the hardware and software inventory information on the NetWare inventoried servers. For more information, review the following sections:

- “Scanning for the Hardware Inventory Information” on page 497
- “Scanning for the Software Inventory Information” on page 497

Scanning for the Hardware Inventory Information

Following are the sources on the NetWare inventoried servers from where the hardware inventory information is scanned:

- “Simple Network Management Protocol (SNMP)” on page 497
- “SMBIOS” on page 497
- “Probe” on page 497

For more information about the hardware information collected by the Inventory scanner, see Appendix I, “Hardware Information Collected by the Inventory Scanners,” on page 689.

Simple Network Management Protocol (SNMP)

The scanners collect certain hardware (device) and network information based on SNMP. Additionally, the scanner also uses SNMP to report software installed and registered in products.dat. The scanner uses SNMP v2.0 and the services of hostmib.nlm,ipxrtr.nlm, and ipxrtrnm.nlm.

SMBIOS

The Inventory scanners use the SMBIOS information embedded into the BIOS of most hardware to report BIOS version, BIOS release date, Manufacturer, asset tag, product name, serial number, processor information, cache information, system slots information, port information, video adapter name, sound card name, and so on.The Inventory scanner reads information from SMBIOS with the help of invaid.nlm.

Probe

Probe is a special built-in algorithm in the Inventory scanner, which is used to collect hardware information.

Scanning for the Software Inventory Information

The Inventory scanner scans the following software inventory information on the NetWare inventoried servers:

- “Installed Software Information” on page 498
- “Disk Usage” on page 499
- “File Information” on page 499
- “AntiVirus Definition Files” on page 499

Installed Software Information

The scanner collects software information from the following sources on the inventoried servers: Microsoft* Installer (MSI), Add-Remove Programs, Dictionary-based scan, and Probe.

Products.Dat: includes software that are installed on the NetWare inventoried servers.

Dictionary-based scan: includes software that are collected based on the software dictionary rules. For more information, see "Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers" on page 550.

Probe: Probe is a special built-in algorithm in the Inventory scanner, which is used to collect software information about ZENworks Suite and its installed components.

The following table shows the software information collected by the scanner from the various sources:

Scanned Attributes	Product.Dat	Dictionary-based scan	Probe
Product Name	Yes	Yes	Yes
Vendor Name	No	Yes	Yes
Product Version	Yes	Yes	Yes
Product Identifier	No	No	No
Product Install Location	No	Yes	Yes
Category	No	Yes	No
Description	No	Yes	No
Help Link	No	No	No
MSI Package GUID	No	No	Yes
Display/Internal Version	No	Yes	Yes
Language	No	No	Yes
UnInstall String	No	No	No
Installation Source	No	No	No
Display Name	No	No	Yes
Support Pack	No	No	Yes
Product Edition	No	No	Yes
Last Execution Time	No	No	No
Usage Count	No	No	No

Disk Usage

The scanner collects the total disk usage information for the file extensions that are configured in the Software Dictionary editor. For more information, see “Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers” on page 550.

File Information

The scanner reports the following information for the files that match with the dictionary entries, and the files that belong to the unidentified software list, which is configured using the software dictionary rules. For more information, see “Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 580

The scanner reports the following file attributes: FileName, FileSize, LastModifiedTime, InternalName, FileVersion, ProductName, ProductVersion, CompanyName, Language, DirectoryPath, and SoftwareDictionaryID.

AntiVirus Definition Files

The scanner collects information about the latest virus definition date and version of McAfee Netshield* 4.6.*x* installed on the inventoried servers.

Scanning for the Windows Inventoried Servers

The Inventory scanner scans for the hardware and software inventory information on the Windows inventoried servers. For more information, review the following sections:

- “Scanning for the Hardware Inventory Information” on page 499
- “Scanning for the Software Inventory Information” on page 501

Scanning for the Hardware Inventory Information

Following are the sources on the inventoried servers from where the hardware inventory information is scanned:

- “Desktop Management Interface (DMI)” on page 499
- “Windows Management Instrumentation (WMI)” on page 500
- “Probe” on page 500

For more information about the hardware information collected by the Inventory scanner, see Appendix I, “Hardware Information Collected by the Inventory Scanners,” on page 689.

Desktop Management Interface (DMI)

The scanners for scanning the inventoried servers also include scanning based on the industry-standard Desktop Management Interface (DMI) specification 2.0. These programs use the Management Interface (MI) of DMI to look for the hardware components installed on the inventoried server. The scanners will scan for specific components that are instrumented on the inventoried server through DMI. The scanners will query the DMI service layer to retrieve this information.

The MI allows the DMI-compliant scanners to probe the Service Provider within the Service Layer. The Service Provider collects information from the manageable components and stores the collected information in the Management Information Format database.

The Component Interface (CI) communicates with the manageable components and the Service layer. The following figure shows the scanner interaction with DMI.

For more information on DMI standards, see the DMTF Web site (http://www.dmtf.org).

NOTE: If the inventoried servers are DMI compliant and if the Enable DMI Scan check box is selected in the Server Inventory policy, the scanners will collect hardware information by querying the DMI Service Layer. Otherwise, the scanners probe for the hardware.

We recommend that you instrument DMI and also install DMI components that are supplied by the vendors.

For example, if you have a Compaq Family Deskpro EN Model-SFF6500 running under Windows 98, download the Management Product software - Compaq Insight Management Desktop Agents software for Windows 98 from the Compaq Web site.

For Dell, access the DM/Desktop Management Utilities software from the Dell Web site.

Windows Management Instrumentation (WMI)

The scanners collect hardware information from Windows inventoried servers based on Microsoft's Windows Management Instrumentation (WMI) specification.

WMI is the Microsoft implementation of Web-Based Enterprise Management (WBEM) that enables accessing management information in an enterprise environment. WMI 1.5 is fully compliant with Common Information Model (CIM) schema, which is an industry standard. For more information, see Microsoft WMI Web site (http://www.microsoft.com/hwdev/driver/WMI). WMI also works with existing management standards, such as DMI and SNMP.

The scanners use WMI to look for the hardware components installed on the inventoried server. The scanners also scan for specific components that are instrumented on the inventoried server through WMI.

WMI-compliant scanners are supported on Windows inventoried servers only.

You can view the WMI information of the inventoried servers in the Server Inventory.

To obtain WMI information from the inventoried server, you must first download Microsoft's Windows Management Instrumentation - Core Software Installation from Microsoft WMI Web site (http://msdn.microsoft.com/downloads/default.asp?url=/downloads/sample.asp?url=/msdn-files/027/001/576/msdncompositedoc.xml), and then install WMI Core Software on Windows 98 servers

IMPORTANT: Only the WMI Core Software Installation download is required to instrument an inventoried server for WMI. To troubleshoot any WMI related problems, you can use the WMI SDK download. Also, on Windows 2000 servers, the WMI Core Software is already installed.

By default, both DMI and WMI scanning will be enabled. To disable either DMI or WMI scanning, deselect the Enable DMI or Enable WMI check box, in the Inventory policy window.

Probe

Probe is a special built-in algorithm in the Inventory scanner, which is used to collect software information.

Scanning for the Software Inventory Information

The Inventory scanner scans for the following software inventory information on the Windows inventoried servers:

- “Installed Software Information” on page 501
- “Disk Usage” on page 502
- “File Information” on page 502
- “AntiVirus Definition Files” on page 502

Installed Software Information

The scanner collects software information from the following sources on the inventoried woks tat ion: Microsoft Installer (MSI), Add-Remove Programs, Dictionary-based scan, and Probe.

MSI: includes software that are installed on the inventoried servers using the Microsoft Installer.

Add-Remove Programs: includes software that are listed in the Add-Remove Programs window.

Dictionary-based scan: includes software that are collected based on the software dictionary rules. For more information, see “Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers” on page 550

Probe: Probe is a special built-in algorithm in the Inventory scanner, which is used to collect software information about Windows operating system, Internet Explorer, Media Player, Outlook Express, Microsoft Office and its installed components, Novell Client and its installed components, and ZENworks suite and its installed components.

The following table shows the software information collected by the scanner from the various sources:

Scanned Attributes	MSI	Add-Remove Programs	Dictionary-based scan	Probe
Product Name	Yes	Yes	Yes	Yes
Vendor Name	Yes	No	Yes	Yes
Product Version	Yes	Yes	Yes	Yes
Product Identifier	Yes	Yes	No	No
Product Install Location	Yes	Yes	Yes	Yes
Category	No	No	Yes	No
Description	No	No	Yes	No
Help Link	Yes	Yes	No	No
MSI Package GUID	Yes	Yes	No	Yes
Display/Internal Version	Yes	Yes	Yes	Yes
Language	Yes	Yes	No	Yes
UnInstall String	Yes	Yes	No	No
Installation Source	Yes	Yes	No	No

Scanned Attributes	MSI	Add-Remove Programs	Dictionary-based scan	Probe
Display Name	Yes	Yes	No	Yes
Support Pack	No	No	No	Yes
Product Edition	No	No	No	Yes
Last Execution Time	No	Yes	No	No
Usage Count	No	Yes	No	No

Disk Usage

The scanner collects the total disk usage information for the file extensions that are configured in the Software Dictionary editor. For more information, see "Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers" on page 550.

File Information

The scanner reports the following information for the files that match with the dictionary entries, and the files that belong to the unidentified software list, which is configured using the software dictionary rules. For more information, see "Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules" on page 580

The scanner reports the following file attributes: FileName, FileSize, LastModifiedTime, InternalName, FileVersion, ProductName, ProductVersion, CompanyName, Language, DirectoryPath, and SoftwareDictionaryID.

AntiVirus Definition Files

The scanner collects information about the latest virus definition date and version that are installed on the inventoried servers for the following product versions:

Symantec AntiVirus Corporate Edition 8.0
Norton AntiVirus Corporate Edition for Windows 7.0
Norton AntiVirus Corporate Edition 7.6.1.0000
Symantec Norton AntiVirus 2000
Symantec Norton Internet Security 2002
Symantec Norton AntiVirus 2003 (9.00)
Symantec Norton AntiVirus 2003 Professional Edition (9.00)
Symantec Norton AntiVirus 2004 (10.00)
Symantec Norton Internet Security 2004 (10.00)
Symantec Norton AntiVirus 2004 Professional (10.00)
Symantec Norton Internet Security 2004 Professional (10.00)
Symantec Norton AntiVirus 2005 Professional (11.00)
Symantec Norton Internet Security 2005 Professional (11.00)
Network Associates McAfee VirusScan 4.0.3 (Windows 9*x*)
Network Associates McAfee VirusScan NT 4.0.3a (Windows NT)
Network Associates McAfee NetShield 4.5.0
Network Associates McAfee VirusScan 4.5.0
Network Associates McAfee VirusScan 4.5.1

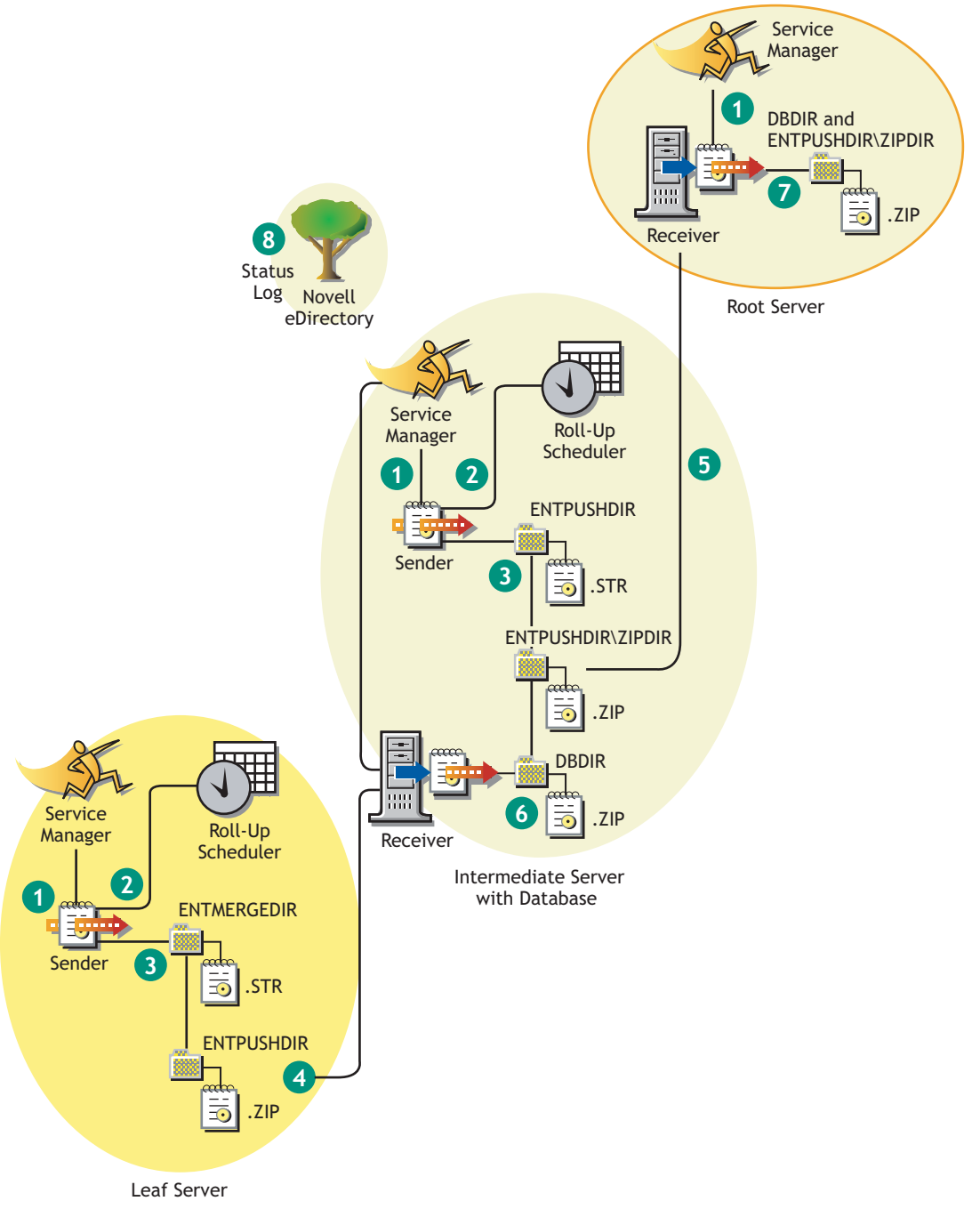
Network Associates McAfee VirusScan (McAfee Security Center) 8.0
Network Associates McAfee VirusScan ASaP
Network Associates McAfee VirusScan Enterprise 7.1
Network Associates McAfee VirusScan Enterprise 8.0
Central Command Vexira AntiVirus Guard for Windows XP (2000 + NT) 2.10
Central Command Vexira AntiVirus Windows 95/98
Central Command Vexira AntiVirus NT/2000 Servers
Central Command Vexira AntiVirus Server Edition (6.26.*xx*.*xx*)
Sophos Anti-Virus - Windows NT/2000/XP/2003
Sophos Anti-Virus - Windows 95/98
Trend Micro PC-cillin 2002 (9.*x*)
Trend Micro PC-cillin 2003 (10.*x*)
Trend Micro Internet Security 11.*x* (PC-cillin)
Trend Micro Internet Security 2005 12.*x* (PC-cillin)
Trend Micro Server Protect 5.*xx*
Trend Micro OfficeScan 5.*xx* - Client for Windows NT/2000/XP
Trend Micro OfficeScan 5.*xx* - Client for Windows 9*x*

Understanding the Sender - Receiver

The Sender and the Receiver on the Inventory servers transfer the scan files from the lower-level Inventory servers to the higher-level Inventory servers. The Sender-Receiver uses the ZENworks Web Server to process the XML-RPC requests. The following sections contain more information:

- “Understanding the Sender” on page 505
- “Understanding the Receiver” on page 506
- “Understanding the Compressed Scan Data File” on page 506
- “Sender-Receiver Directories” on page 507

The following illustration depicts the processing done by the Sender-Receiver.



The processing done by the Sender-Receiver is as follows:

1. The Service Manager starts the Sender-Receiver component.
2. The Roll-Up Scheduler activates the Sender at the specified roll-up time.
3. The Sender moves the scan data files (.str) from the enterprise merge directory (entmergedir) to the enterprise push directory (entpushdir) and compresses the files as a .zip file.
4. Each .zip file is again compressed with the .prp file into a .zip file. The .prp file is an internal file containing information about the .zip file.

5. The Sender sends the .zip file from the \entpushdir directory to the Receiver on the next-level Inventory server.
6. The Receiver places the .zip files to the \entpushdir\zipdir directory.
7. The Receiver copies the .zip files to the \entpushdir directory and deletes the .zip files from the entpushdir\zipdir directory.
8. The Receiver copies the .zip files to the database directory (dbdir) if a database is attached to the Inventory server.
9. The Sender-Receiver logs the status in eDirectory.

Understanding the Sender

The Sender is a Java* component that runs on any Leaf Server or on the Intermediate Server. The Sender is a service loaded by the Service Manager. See “An Overview of the Inventory Components on the Inventory Server” on page 510 for a quick reference table of Inventory server components.

The flow of information from the Sender in the roll-up of inventory information is as follows:

1. The Service Manager starts the Sender on the Inventory server. At the specified time scheduled in the Roll-Up Schedule, the Sender moves the scan data files (.str) from the enterprise merge directory (entmergedir) to the enterprise push directory (entpushdir).

   The Sender compresses these .str files in the \entpushdir directory of the Inventory server as a .zip file and then deletes the .str files. This .zip file is again compressed with the .prp file into a .zip file. The .prp file is an internal file containing information about the .zip file. For more information, see “Understanding the Compressed Scan Data File” on page 506.
2. The Sender creates a new record in the zeninvRollUpLog attribute of the Inventory Service object in eDirectory with the following details: server on which the Sender compresses the .str files and the name and size of the .zip file.
3. Based on the Discard Scan Data Time in the Inventory Service object properties of the Receiver, the Sender deletes the compressed .zip files in the \entpushdir directory that have been created earlier than the specified discard scan data time. This removes unwanted scan information being sent in the roll-up.
4. The Sender sends the compressed .zip files to the Receiver, with the oldest compressed files sent first.
5. The Sender after transferring the .zip file, deletes the compressed files in the \entpushdir directory.
6. After the roll-up of information, the Sender updates the zeninvRollUpLog attribute of the Inventory server on which the compressed file was created with the following details: Inventory server from which the Sender transmitted the file, name of the .zip file, time of transmission, total time taken to transmit the files, and the Inventory server to which it was sent.

   In case of rolling up inventory information across trees, the roll-up status messages are logged into the first inventory server receiving the .zip file in the tree.

   The status information for all actions of the Sender is logged in the Roll-Up Log and Server Status log. For more information, see “Monitoring Server Inventory Using Status Logs” on page 677.

If the Sender is unable to connect to the Receiver, the Sender retries to connect after 10 seconds. The time interval increases exponentially by a factor of 2. After 14 retries, the Sender stops trying to connect to the Receiver. The Sender retries for approximately 23 hours before it discontinues trying. The Sender does not process any other information while it is establishing the connection.

Understanding the Receiver

The Receiver is a Java component that runs on the Intermediate Server or on the Root Server. The Receiver is a service loaded by the Service Manager. See "An Overview of the Inventory Components on the Inventory Server" on page 510 for a quick reference table of Inventory server components.

The processing done by the Receiver is as follows:

1. The Receiver receives the scan .zip file from the Sender and places the file in the \entpushdir\zipdir directory.
2. The Receiver copies the .zip file to the \entpushdir directory and deletes the .zip files from the \entpushdir\zipdir directory.

   On an Intermediate Server, the file is placed in \entpushdir. On an Intermediate Server with Database, or an Intermediate Server with Database and Inventoried Servers, the file is placed in \entpushdir and copied to the Database Directory (dbdir).
3. The Receiver on the Root Server or the Root Server with Inventoried Servers receives the .zip files from the Senders and places the .zip files in the \entpushdir\zipdir directory. It copies the files to the \dbdir directory on the Inventory server.
4. The Receiver logs the status information in the Roll-Up log. For more information, see "Monitoring Server Inventory Using Status Logs" on page 677.

Understanding the Compressed Scan Data File

The Sender compresses the scan data files (.str) into a .zip file. This .zip file is again compressed with the .prp file into a .zip file. The .zip file (containing the .zip files and .prp) is named using the following naming conventions:

scheduledtime_inventoryservername_treename_storedstatus.zip

where *scheduledtime* refers to the date and time when the .zip file is created, *inventoryservername* refers to the Inventory server on which the .zip file was compressed, *treename* refers to the unique tree name in which the .zip file is currently located, *storedstatus* refers to the storage status of the .zip file, and *ZIP* is the file extension for the compressed files.

The *storedstatus* is represented by 0, 1, or 2. 0 indicates the .zip file has not yet been stored. 1 indicates the .zip file will be stored for the first time in the Inventory server. 2 indicates the .zip file has already been stored once.

The .zip filename changes depending on if the database is attached to the Inventory server.

The .zip file contains the .zip files and a property file. The property file is named using the following conventions:

scheduledtime_inventoryservername.prp

The property file contains the scheduled time, Inventory server name, and signature. The signature helps to authenticate the .zip file. Each .zip file can contain a maximum of 50 .str files

Sender-Receiver Directories

The following table provides a quick reference of the directories that the Sender-Receiver uses:

Server	Sender	Receiver	Entmergdir	Entpushdir \ Zipdir	Entpushdir	Dbdir
Leaf Server, Leaf Server with Database	Runs on this Inventory server	--	Sender moves the .str files to \entpushdir.	--	Sender compresses the .str files as a .zip file. Sender deletes the .str files. Sends the .zip file to the next-level Inventory server.	--
Intermediate Server	Runs on this Inventory server	Runs on this Inventory server	--	Receiver receives the .zip files from the lower-level Inventory server in this directory.	Receiver copies the .zip files from the lower-level Inventory server in this directory. Sender sends the .zip files to the next-level Inventory server.	--
Intermediate Server with Inventoried Servers	Runs on this Inventory server	Runs on this Inventory server	Sender moves the .str files to \entpushdir.	Receiver receives the .zip files from the lower-level Inventory server in this directory.	Receiver copies the .zip files from \zipdir into this directory. Sender sends the .zip files to the next-level Inventory server. Sender compresses the .str files in to .zip files. Sender deletes the .str files.	--
Intermediate Server with Database	Runs on this Inventory server	Runs on this Inventory server	--	Receiver receives the .zip files from the lower-level Inventory server in this directory.	Receiver copies the .zip files from \zipdir into this directory. Sender sends the .zip file to the next-level Inventory server.	Receiver copies the file in this directory.
Intermediate Server with Database and Inventoried Servers	Runs on this Inventory server	Runs on this Inventory server	Sender moves the .str files to \entpushdir.	Receiver receives the .zip files from the lower-level Inventory server in this directory.	Receiver copies the .zip files from \zipdir into this directory. Sender compresses the .str files as a .zip file. Sender deletes the .str files. Sender sends the .zip file to the next-level Inventory server.	Receiver copies the file in this directory.
Root Server, Root Server with Inventoried Servers	--	Runs on this Inventory server	--	Receiver receives the .zip files from the lower-level Inventory server in this directory.	--	Receiver copies the .zip files from the lower-level Inventory server in this directory.

On the Standalone Server, the Receiver is not loaded.

Understanding the Selector

The Selector is a Java component on the Inventory server that receives the inventory information from the Inventory servers. These Inventory servers can be any of the following: Leaf Server, Leaf Server with Database, Intermediate Server with Database and Inventoried Servers, Intermediate Server with Inventoried Servers, Root Server with Inventoried Servers, and Standalone Server. See "An Overview of the Inventory Components on the Inventory Server" on page 510 for a quick reference table of Inventory server components.

The processing done by the Selector is as follows:

1. While scanning the inventoried server, the Scanner creates a scan data file (.str) in the scan directory (scandir) at the Inventory server for each scan done on the inventoried server. The location of \scandir is obtained from the Inventory Service object. The Selector processes the .str files placed by the Scan Collector in the \scandir directory.
2. The Selector checks for the following conditions to ensure that the .str file generated by the Scanner is valid:
   - The integer value that is generated by using the .str file and logged into the .str file by the Scanner and the integer value generated by using the .str file by the Selector should be the same.
   - The actual size of the .str file should be in sync with the size recorded in the .str file.

   The Selector processes only valid .str files. If invalid files are present in the directory, the Selector deletes the files.
3. Based on the role of the Inventory server, the Selector copies the .str files to the \dbdir directory (if the database is attached) and the \entmerge directory. If the .str file already exists in the directory, it overwrites the file.

   The following table lists the directories that the Selector copies or renames the files to:

Server	Copies the .str file to the Database Directory (dbdir)	Renames the .str file in the Database Directory (dbdir)	Renames the .str file in the Enterprise Merge Directory (entmergedir)
Leaf Server with Database	Yes	--	Yes
Leaf Server	--	--	Yes
Intermediate Server with Database and Inventoried Servers	Yes	--	Yes
Standalone Server	--	Yes	--
Root Server with Inventoried Servers	--	Yes	--

4. The Selector logs the status in the Server log. For more information, see "Monitoring Server Inventory Using Status Logs" on page 677.

Understanding the Storer

The Storer is a Java component on the Inventory server that has a database attached to it. These Inventory servers can be any of the following: Leaf Server with Database, Intermediate Server with Database, Intermediate Server with Database and Inventoried Servers, Root Server, and Root Server with Inventoried Servers. See “An Overview of the Inventory Components on the Inventory Server” on page 510 for a quick reference table of Inventory server components.

The Storer runs as a Service loaded by the Service Manager. It processes the files in the \dbdir directory.

The processing done by the Storer is as follows:

1. The Storer reads the Startup configuration parameters from the Inventory server Configuration Service.
2. From the Inventory server configuration information stored in eDirectory, the Storer looks in the database directory (dbdir) for the scan files. The Inventory server configuration information determines the location of \dbdir and the database server from the eDirectory policy. The Selector places the .str files in \dbdir and the Receiver places the .zip files in \dbdir.
3. The Storer processes the .str files and the .zip files alternately.
4. The Storer extracts the .zip file containing the compressed .str files and the .prp file to a temp directory (\dbdir\temp) and updates the database with the inventory information of the .str files for the inventoried servers.
5. The Storer updates the status in the Inventory server Status log and updates the Roll-Up log.You can view the Inventory server status information in the Inventory server Status log.

   In case of rolling up inventory information across trees, the roll-up status messages are logged into the first inventory server receiving the .zip file in the tree. For more information, see “Monitoring Server Inventory Using Status Logs” on page 677.

Understanding the Dictionary Provider - Dictionary Consumer

All Inventory servers and inventoried servers have Dictionary Provider and Dictionary Consumer services that are automatically installed during the Server Inventory installation.

When an inventoried server is started, the Dictionary Consumer reads the Dictionary Update policy and contacts the Dictionary Provider (running on the Inventory server) specified in the policy.

Subsequently, the Dictionary Consumer checks for the dictionary updates based on the schedule set in the Dictionary Update policy. It compares the date of the dictionary file on the Dictionary Provider with the file that has been locally stored. If the file on the Dictionary Provider is latest, then the Dictionary Consumer downloads the file from the Dictionary Provider using XML-RPC as per the schedule.

Understanding the Upgrade Service

The Upgrade service runs as a service loaded by the Service Manager and performs the following functions:

1. Migrates ZENworks for Servers 3.0.2 database to ZENworks 6.5 Server Management database.
2. Converts the ZENworks for Servers 3.0.2 residue .str files to ZENworks 6.5 Server Management .str files.

The Upgrade service corrects the Inventory database schema and information to make it compatible with ZENworks 6.5 Server Management and ZENworks 6.5 Desktop Management. The Upgrade service performs all the functions in a state-driven method. This is to make sure that the Upgrade service does not execute the same steps when one step is executed successfully. The Upgrade service runs as an uninterrupted service. Therefore, you cannot manually stop the Upgrade service. The Upgrade service stops automatically after completing all its functions.

The Database migration activity is additionally traced into a migration log, which could be found in the *installation_path*\zenworks\inv\server\wminv\logs\migrationlogs directory.

An Overview of the Inventory Components on the Inventory Server

Depending on the type of the Inventory server, the following inventory components exist on the Inventory server.

Server Component	Root Server	Root Server with Inventoried Servers	Leaf Server	Leaf Server with Database	Intermediate Server	Intermediate Server with Database and Inventoried Servers	Intermediate Server with Database	Intermediate Server with Inventoried Servers	Standalone Server
Service Manager	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Scan Collector	No	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Selector	No	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Storer	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes
Sender	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Receiver	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No
Database	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes
Inventory Removal Service	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Upgrade Service	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes

Server Component	Root Server	Root Server with Inventoried Servers	Leaf Server	Leaf Server with Database	Intermediate Server	Intermediate Server with Database and Inventoried Servers	Intermediate Server with Database	Intermediate Server with Inventoried Servers	Standalone Server
Dictionary Consumer and Dictionary Provider	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Understanding the Inventory Database

Server Inventory provides a centralized Common Information Model (CIM)-compliant Sybase database. The Inventory database serves as a repository of hardware and software information for the servers. The network administrator can view the inventory information, query the database, and generate inventory reports in ConsoleOne. For more information, see Chapter 15, “Understanding the ZENworks 6.5 Server Managements Inventory Database Schema,” on page 513

15 Understanding the ZENworks 6.5 Server Managements Inventory Database Schema

This section describes the design of the Novell® ZENworks® 6.5 Server Management Inventory database schema implemented using the Common Information Model (CIM) of Distributed Management Task Force (DMTF). To understand this section effectively, you should be familiar with terminology such as CIM and Desktop Management Interface (DMI). You should also have a solid understanding of Relational Database Based Managed Systems (RDBMS) and database concepts.

The following sections provide in-depth information:

Overview

The DMTF is the industry organization leading the development, adoption, and unification of management standards and initiatives for desktop, enterprise, and Internet environments. For more information about DMTF, see the DMTF Web site (http://www.dmtf.org).

The DMTF CIM is an approach to system and network management that applies the basic structuring and conceptualization techniques of the object-oriented paradigm. The approach uses a uniform modeling formalism that together with the basic repertoire of object-oriented constructs supports the cooperative development of an object-oriented schema across multiple organizations.

A management schema is provided to establish a common conceptual framework at the level of a fundamental topology, both with respect to classification and association, and to a basic set of classes intended to establish a common framework for a description of the managed environment. The management schema is divided into the following conceptual layers:

- **Core Model:** An information model that captures notions that are applicable to all areas of management.
- **Common Model:** An information model that captures notions that are common to particular management areas, but independent of a particular technology or implementation. The common areas are systems, applications, databases, networks, and devices. The information model is specific enough to provide a basis for the development of management applications. This model provides a set of base classes for extension into the area of technology-specific schema. The Core and Common models together are expressed as the CIM schema.
- **Extension Schema:** This schema represents technology-specific extensions of the Common model. These schema are specific to environments, such as operating systems, for example, NetWare® or Microsoft* Windows*.

CIM comprises a specification and a schema (see the DMTF Web site (http://www.dmtf.org/standards/standard_cim.php). The specification defines the meta-schema plus a concrete representation language called Managed Object Format (MOF).

CIM Schema

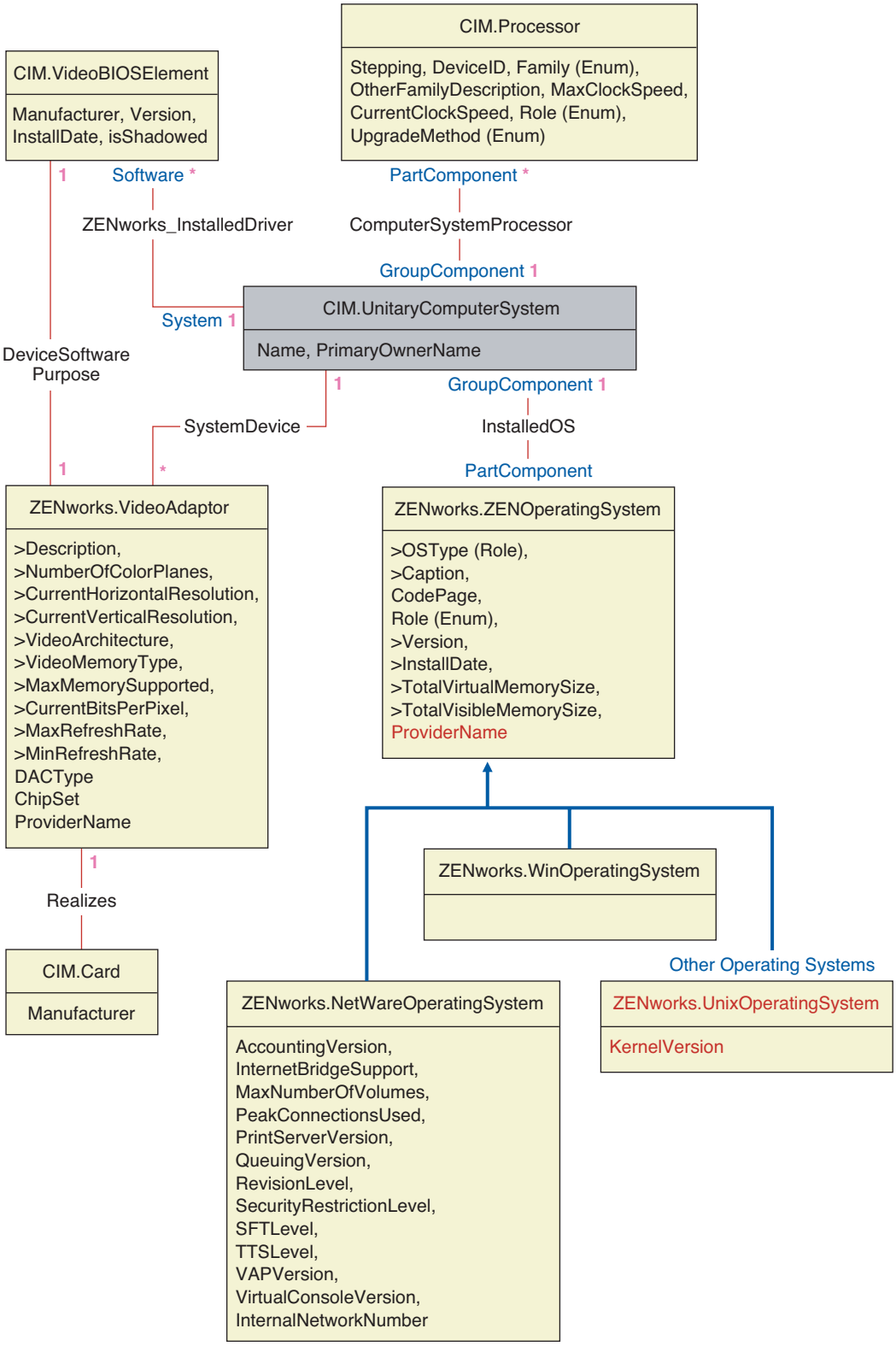
The elements of the meta schema are classes, properties, and methods. The meta schema also supports indications and associations as types of classes and references as types of properties.

Classes can be arranged in a generalization hierarchy that represents subtype relationships between classes. The generalization hierarchy is a rooted, directed graph that does not support multiple inheritance.

A regular class may contain scalar or array properties of any intrinsic type such as Boolean, integer, string, and others. It cannot contain embedded classes or references to other classes.

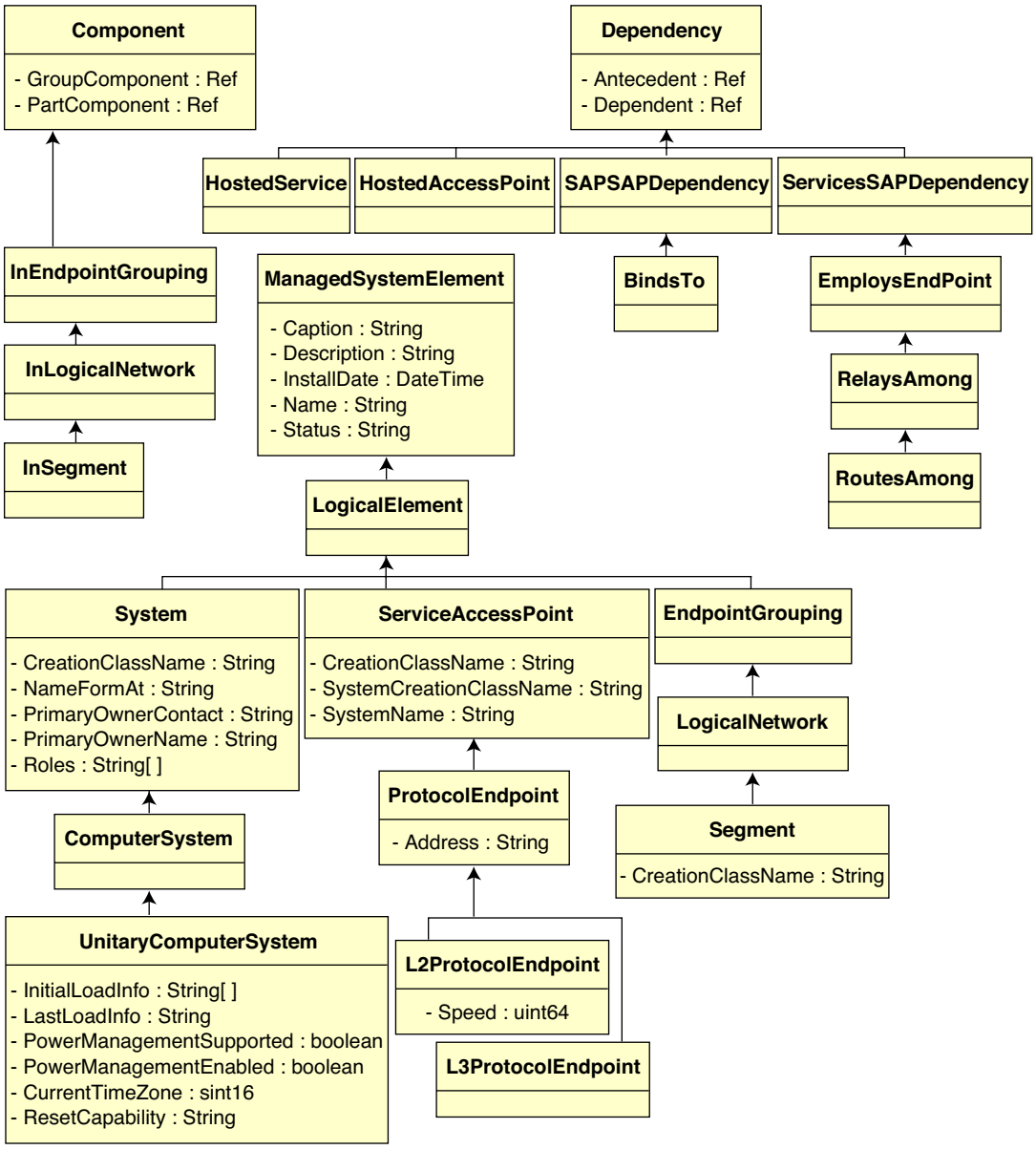
An association is a special class that contains two or more references. It represents a relationship between two or more objects. Because of the way associations are defined, it is possible to establish a relationship between classes without affecting any of the related classes. That is, addition of an association does not affect the interface of the related classes. Only associations can have references.

The schema fragment in the following illustration shows the relationships between some CIM objects that ZENworks 6.5 Server Management uses.



The illustration shows how the CIM schema maps to a relational DBMS schema. The classes are shown with the class name as the box heading. The associations are labeled within the lines between two classes.

The inheritance hierarchy of this schema fragment is shown in the following illustration of the CIM 2.2 schema. The references shown as type Ref are in bold with each association sub-type narrowing the type of the reference.



CIM-to-Relational Mapping

CIM is an object model complete with classes, inheritance, and polymorphism. The generated mapping to a relational schema preserves these features to the maximum extent. The following two aspects are part of the relational mapping:

- **Logical Schema:** The logical schema defines how the information appears to applications, similar to an API. The goal is that the logical schema remains the same irrespective of the underlying database so that application software can run unchanged on any supported databases. Though SQL (pronounced as sequel) is a standard, this goal is not fully possible. Application software will need to know more about the database in use and this information can be abstracted and isolated to a small area of the application code.

- **Physical Schema:** The physical schema defines how the information is structured in the database. The schema tends to be specific to the database because of the nature of SQL and RDBMS. This document will describe the physical schema in general terms only.

A table in the database represents each class in the CIM hierarchy. A column of the appropriate type in the table represents each non-inherited property in the class. Each table also has a primary key, id$, which is a 64-bit integer that uniquely identifies an instance. An instance of a CIM class is represented by a row in each table that corresponds to a class in its inheritance hierarchy. Each row has the same value for id$.

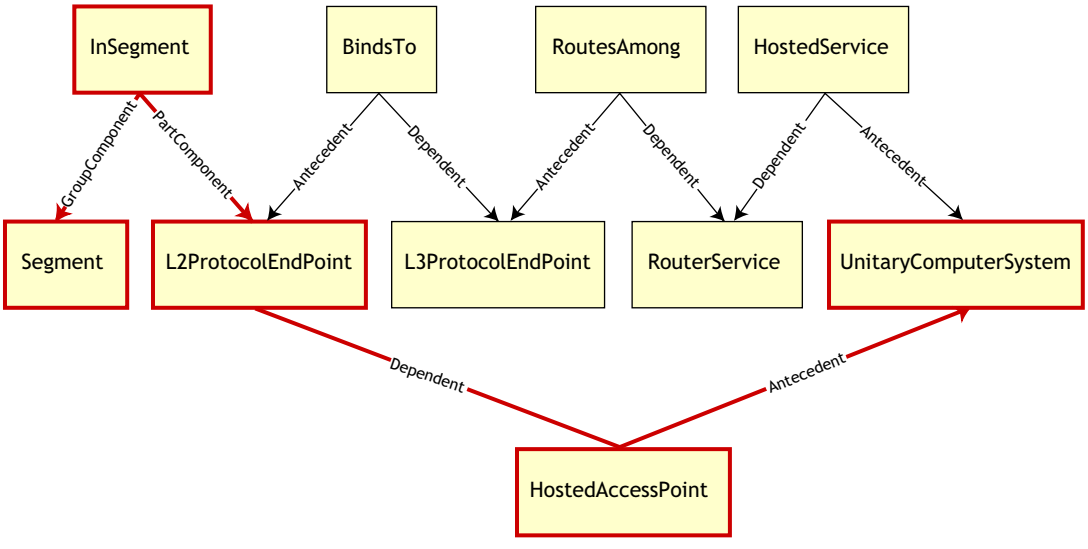
Each CIM class is also represented by a view that uses id$ to join rows from the various tables in the inheritance hierarchy to yield a composite set of properties (inherited plus local) for an instance of that class. The view also contains an extra column, class$, of type integer that represents the type of the actual (leaf-most) class of the instance.

Associations are mapped in the same manner as regular classes, with a reference property being represented by a column with the id$ field of the referenced object instance. Thus, associations can be traversed by doing a join between the reference field in the association and the id$ field in the referenced table.

The following illustration depicts a typical query using this mapping:

```
Get Computers for Segment

SELECT CIM.UnitaryComputerSystem.*
FROM    CIM.UnitaryComputerSystem, CIM.Segment, CIM.L2ProtocolEndPoint,
          CIM.HostedAccessPoint, CIM.InSegment
WHERE  CIM.SegmentName = 'xxx'
AND     CIM.InSegment.GroupComponent = CIM.Segment.id$
AND     CIM.InSegment.PartComponent = CIM.L2ProtocolEndPoint.id$
AND     CIM.HostedAccessPoint.Dependent = CIM.L2ProtocolEndPoint.id$
AND     CIM.HostedAccessPoint.Antecedent = CIM.UnitaryComputerSystem.id$
```



This query finds all the computers attached to a given network segment. The classes and relationships involved are highlighted with borders.

The following topics describe both the schema types:

- “Logical Schema” on page 518
- “Physical Schema” on page 520

Logical Schema

The logical schema is the database schema as seen by users of the database and the application program. The schema consists of stored procedures and views. The underlying tables are not visible to the application.

Inventory components use JDBC to issue SQL statements to the RDBMS and to convert between RDBMS data types and Java* data types. The use of JDBC with stored procedures and views provides a level of abstraction that insulates application code from the underlying database technology and from changes to the physical schema.

The various elements of the logical schema are discussed in more detail in the following sections:

- “Naming Schema Elements” on page 518
- “Users and Roles” on page 519
- “Data Types” on page 519
- “Views” on page 519

Naming Schema Elements

We recommend that you use the CIM names unchanged in the database schema. Some problems may still ensue because of the differences in the naming schemes, such as the following:

- Names in CIM and SQL are not case sensitive.
- All databases have different sets of reserved words that must be enclosed in quotes (“ ”) when used as schema element names; however, in Oracle*, enclosing a name in quotes makes it case sensitive.
- CIM classes avoid using SQL reserved words as names.
- CIM names are not limited in length and usually the names are long. Sybase allows up to 128 characters, but Oracle restricts the names to 30 characters.

Most of these problems are avoided during schema generation by preserving the case of CIM names, abbreviating any names longer than 30 characters, and placing quotes around any name that is in the union of the sets of reserved words.

Any name longer than 28 characters is abbreviated to a root name of 28 or fewer characters to allow a two-character prefix so that all associated SQL schema elements can use the same root name. The abbreviation algorithm shortens a name so that it is mnemonic, recognizable, and also unique within its scope. The abbreviated name is given a # character as a suffix (note that # is an illegal character in CIM) to prevent clashes with other names. If two or more names within the same scope generate the same abbreviation, an additional digit is appended to make the name unique. For example, AttributeCachingForRegularFilesMin is abbreviated to AttCacForRegularFilesMin#.

All such mangled names are written to the mangled name table so that a program can look up the real CIM name and retrieve the mangled name to use with the SQL.

Views are the schema elements that are most often manipulated by application code and queries. They use the same name as the CIM class they represent. For example, the CIM.UnitaryComputerSystem class is represented by a view named CIM.UnitaryComputerSystem.

When necessary, names for indexes and auxiliary tables are created by concatenating the class name and property name separated by a $ character. These names are usually abbreviated. For example, NetworkAdapter$NetworkAddresses is abbreviated to NetAdapter$NetAddresses#. This does not have any adverse impact on ZENworks 6.5 Server Management schema users.

Users and Roles

In SQL, a user with the same name as the schema is the owner of each schema, for example, CIM, ManageWise®, ZENworks®, and others.

Additionally, there is an MW_DBA user that has Database Administrator privileges and rights to all schema objects. The MW_Reader role has read-only access to all schema objects and the MW_Updater role has read-write-execute access to all schema objects.

Application programs should access the database as either MW_Reader or MW_Updater for a Sybase database, MWO_Reader or MWO_Updater for an Oracle database, and MWM_Reader or MWM_Updater for MS SQL Server database depending on their requirements.

Data Types

CIM data types are mapped to the most appropriate data type provided by the database. Usually, the Java application does not require the type because it uses JDBC to access the data.

Java does not natively support unsigned types, so you should use classes or integer types of the next size to represent them. Also, ensure that there are no problems while reading or writing to the database. For example, reading or writing a negative number to an unsigned field in the database is likely cause an error.

Strings in CIM and Java are Unicode*, so the database is created using the UTF8 character set. Internationalization does not pose any problems; however, it may create problem with case sensitivity in queries.

All databases preserve the case of string data stored within them, but may access the data as either case sensitive or otherwise during queries. In ZENworks 6.5 Server Management, the Inventory Query component is not affected because the queried data is retrieved from the database before being queried and so case sensitivity is automatically taken care of.

In CIM, strings may be specified with or without a maximum size in characters. Many strings have no specified size, which means they can be unlimited in size. For efficiency reasons, these unlimited strings are mapped to a variable string with maximum size of 254 characters. CIM strings with a maximum size are mapped to variable database strings of the same size. The size in the database is in bytes and not as characters because a Unicode character may require more than one byte for storage.

Views

Each CIM class is represented in the database by a view that contains all the local and inherited non-array properties of that class. The view is named the same as the CIM class.

Views can be queried using the SELECT statement and updated using the UPDATE statement. Because views cannot be used with the INSERT and DELETE statements, use the constructor and destructor procedures.

- Table definitions 't$*xxx*'
- Index definitions 'i$*xxx*'
- Trigger definitions 'x$*xxx*', 'n$*xxx*' and 'u$*xxx*'
- Sequence definitions (Oracle) 's$*xxx*'
- Stored procedures and functions

The logical schema is layered on top of the physical schema and makes it unnecessary for users and applications to know the physical schema.

Inventory Database Schema in ZENworks 6.5 Server Management

The following section describes the database schema classes and the extensions and associations made to the CIM schema for use in ZENworks 6.5 Server Management. These extensions have ZENworks or ManageWise as their schema name. ZENworks.*classname* refers to the extended class in the ZENworks schema and ManageWise.*classname* refers to the extended class in the ManageWise schema.

The following sections will help you understand the ZENworks 6.5 Server Management database schema:

- “Case Study of CIM Schema Implementation in ZENworks 6.5 Server Management” on page 520
- “Legends for Schema Diagrams” on page 523
- “Schema Diagrams of CIM and the Extension Schema in ZENworks 6.5 Server Management” on page 523
- “Software Inventory Schema” on page 531
- “Sample Inventory Database Queries” on page 536

Case Study of CIM Schema Implementation in ZENworks 6.5 Server Management

The following scenario describes an inventoried server that has two parallel ports with a specified interrupt number.

In the following schema diagram, the CIM.UnitaryComputerSystem represents a managed inventory system.

In this illustration, class CIM.PointingDevice associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemDevice.PartComponent pointing to CIM.PointingDevice. The relationship between the two classes is one to many. This means a computer system might have more than one pointing devices.

Class CIM.IRQ associates to CIM.PointingDevice using the association CIM.AllocatedResource. Dependent pointing to CIM.PointingDevice and Antecedent pointing to CIM.IRQ.

Class ZENworks.ZENKeyboard associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemDevice.PartComponent pointing to ZENworks.ZENKeyboard. The relationship between the two classes is one to one. This means a computer system can have only one Keyboard.

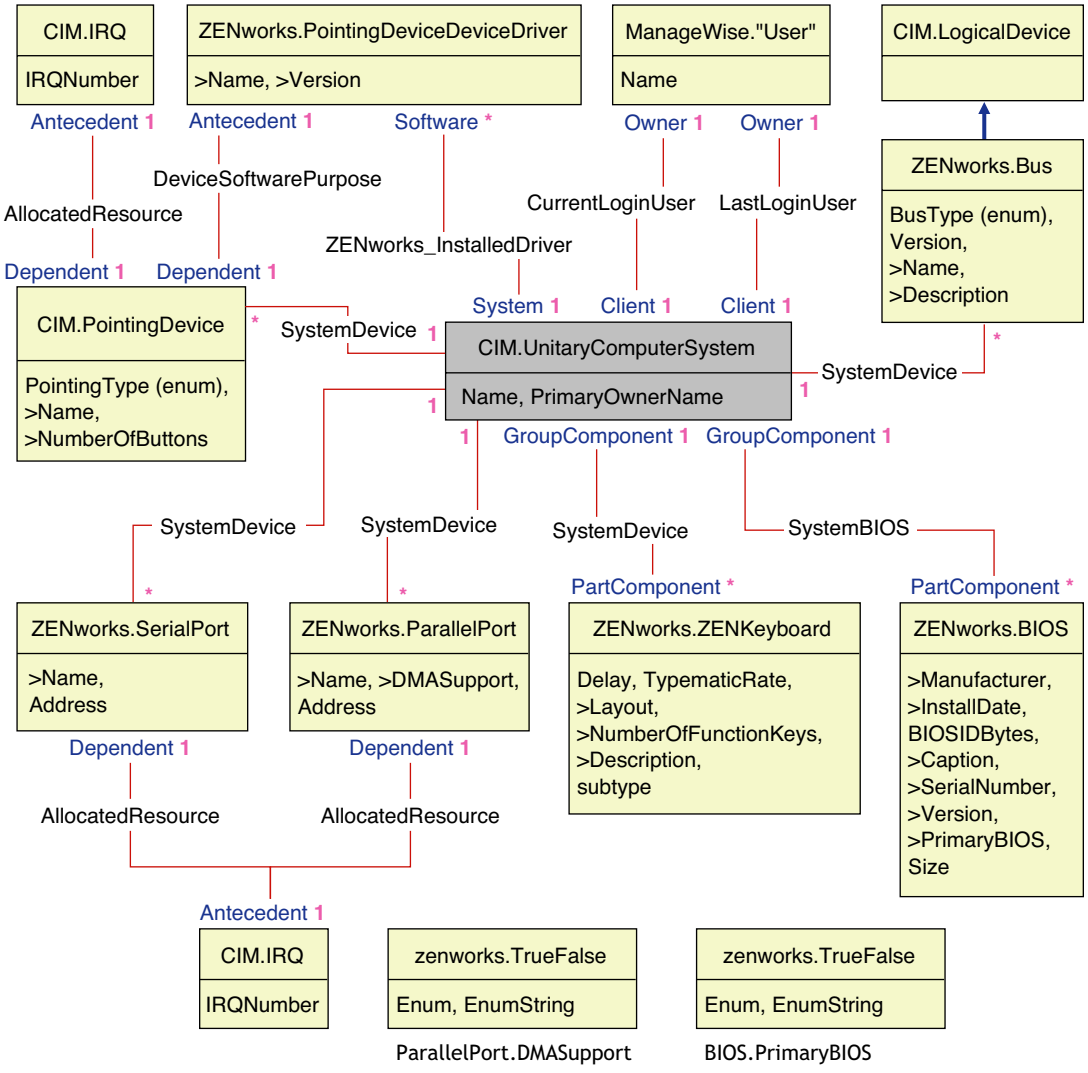
Class ZENworks.BIOS associates to CIM.UnitaryComputerSystem using the association CIM.SystemBIOS with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBIOS.PartComponent pointing to ZENworks.BIOS. The relationship between the two classes is one to one. This means a computer system can have only one BIOS.

Class CIM.ZENworks.ParallelPort associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemDevice.PartComponent pointing to CIM.ZENworks.ParallelPort. The relationship between the two classes is one to many. This means a computer system might have more than one parallel port.

Class ZENworks.BUS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemDevice.PartComponent pointing to ZENworks.BUS. The relationship between the two classes is one to many. This means a computer system can have more than one bus.

Class ManageWise.User associates to CIM.UnitaryComputerSystem using CurrentLoginUser and LastLoginUser. In the CurrentLoginUser association, the specific instance of User is the one who is currently logged into the inventoried server. In the LastLoginUser association, the specific instance of User is the one who logged last into the inventoried server.

Class CIM.IRQ associates to CIM.ParallelPort using the association CIM.AllocatedResource. Dependent pointing to CIM.ParallelPort and Antecedent pointing to CIM.IRQ.



The schema diagram illustrates the following:

- All components that a computer system manages are represented as associations from the UnitaryComputerSystem class. The type of references (1..n, 1..1) between two classes are marked.
- Those associations that do not have a schema name are assumed as CIM schema.

  There are three instances of ZENworks.ParallelPort associated to one instance of CIM.UnitaryComputerSystem using three instances of CIM.SystemDevice associations. CIM.SystemDevice.GroupComponent references UnitaryComputerSystem and CIM.SystemDevice.PartComponent references ParallelPort.

  This is called 1 to n object reference relationship and is depicted in the illustration as 1..*. Similarly, every instance of ParallelPort has a corresponding instance of CIM.IRQ designating the port's irq. This is one-to-one relationship and is depicted as 1..1.

All other classes follow similar representation.

For schema diagrams of other classes, see "Schema Diagrams of CIM and the Extension Schema in ZENworks 6.5 Server Management" on page 523.

Legends for Schema Diagrams

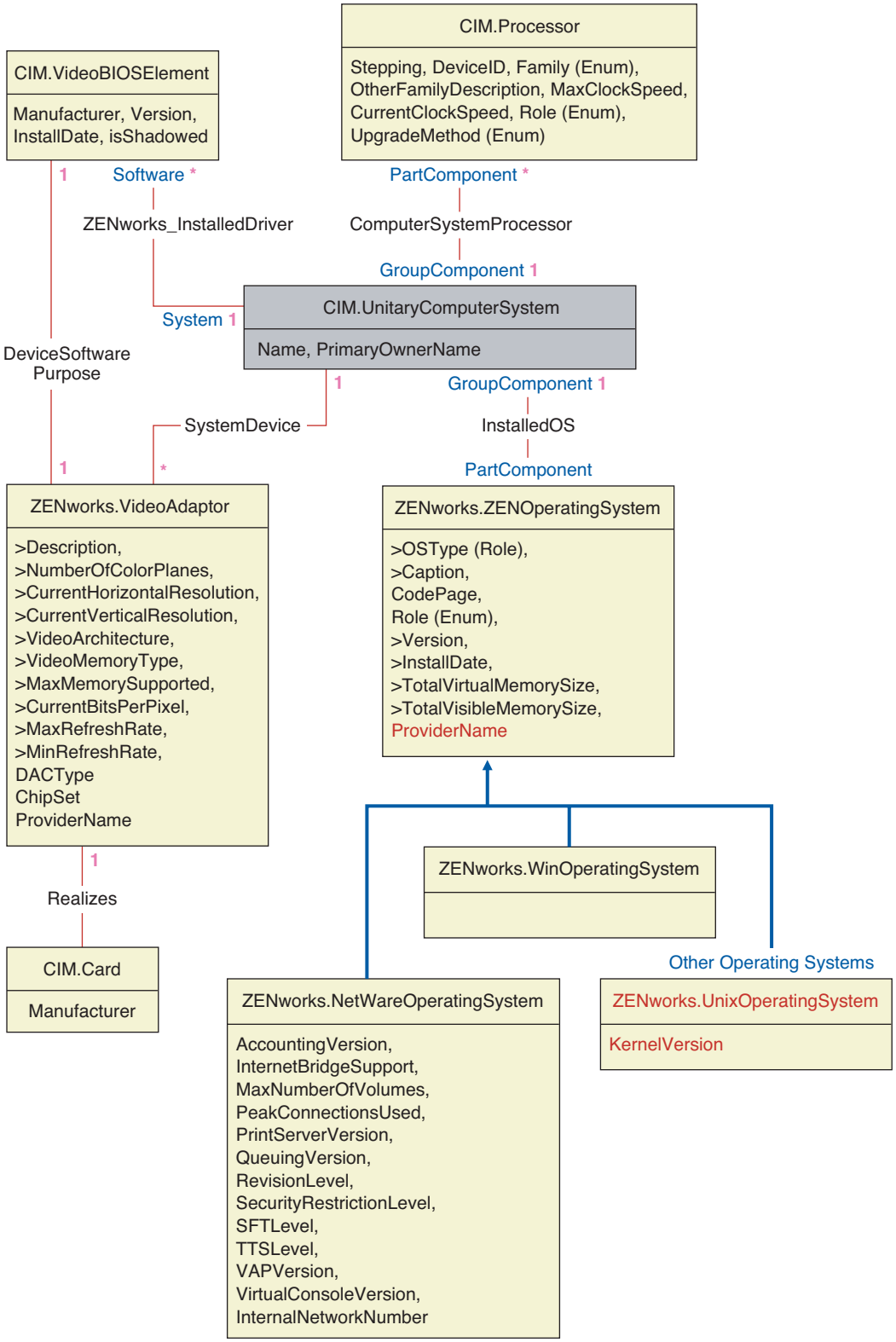
The legends for reading the schema diagrams are as follows:

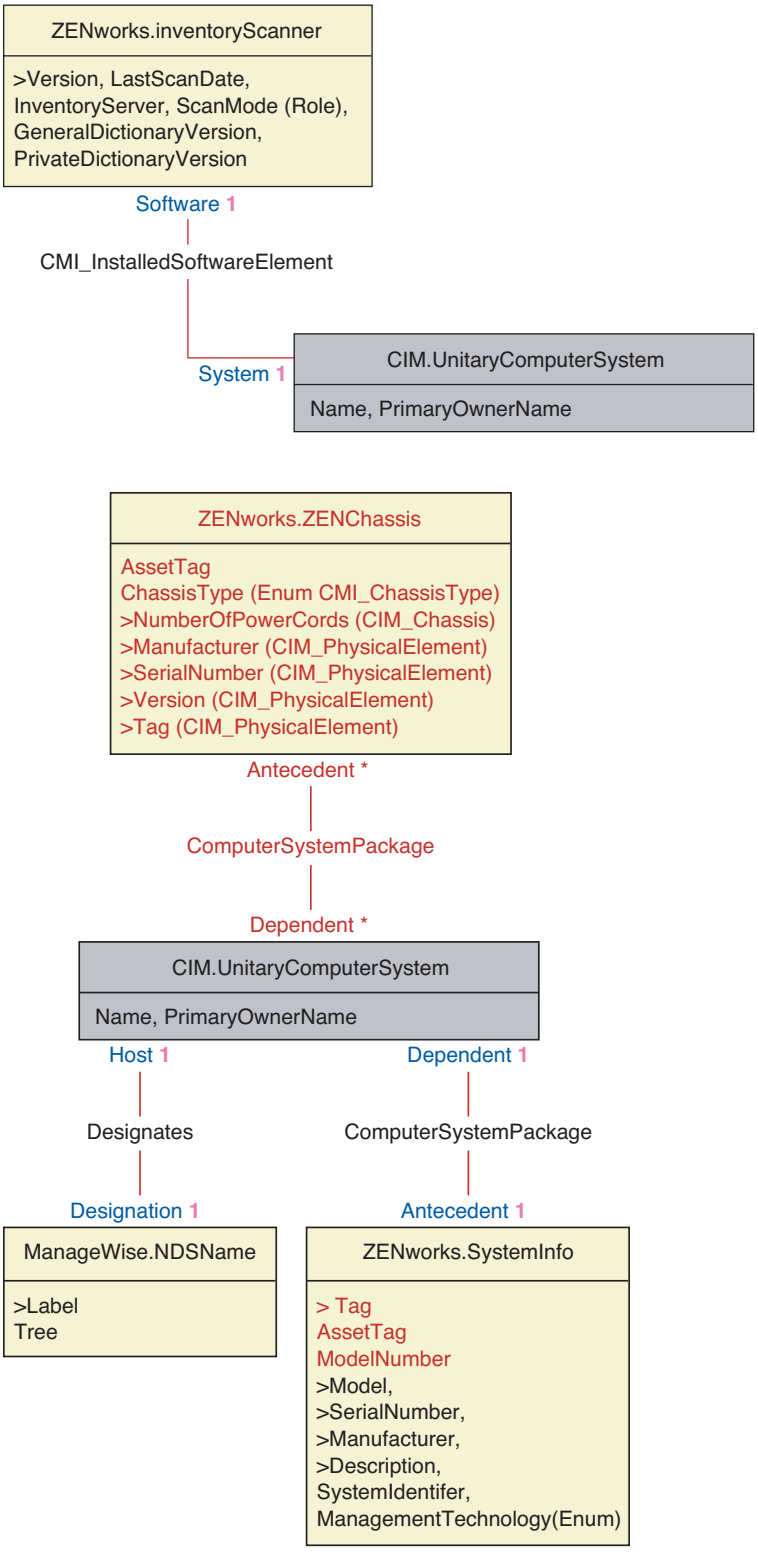
- Class names are enclosed in boxes with the class name as the heading and the attribute names within it.
- Red lines connect two classes using an association class.
- Blue lines indicate the class inheritance hierarchy. The class pointed by the arrow is the class that is being inherited from. The class from where the arrow emanates is the inheriting class.
- The association class name is shown within the line joining two classes.
- References of the association class are marked on either side of the associated classes.

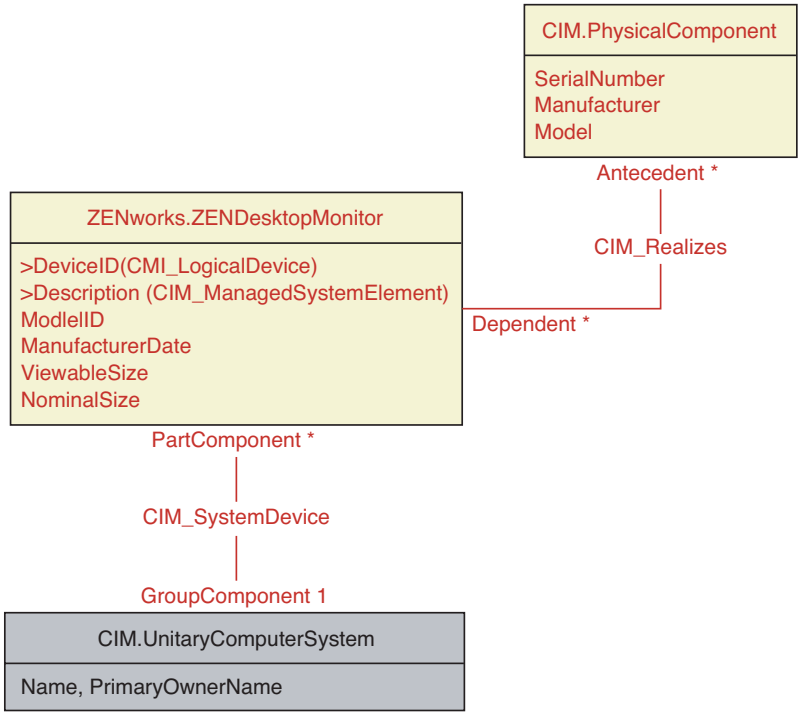
For an explanation about CIM schema, see the CIM 2.2 schema specification on the DMTF Web site (http://www.dmtf.org).

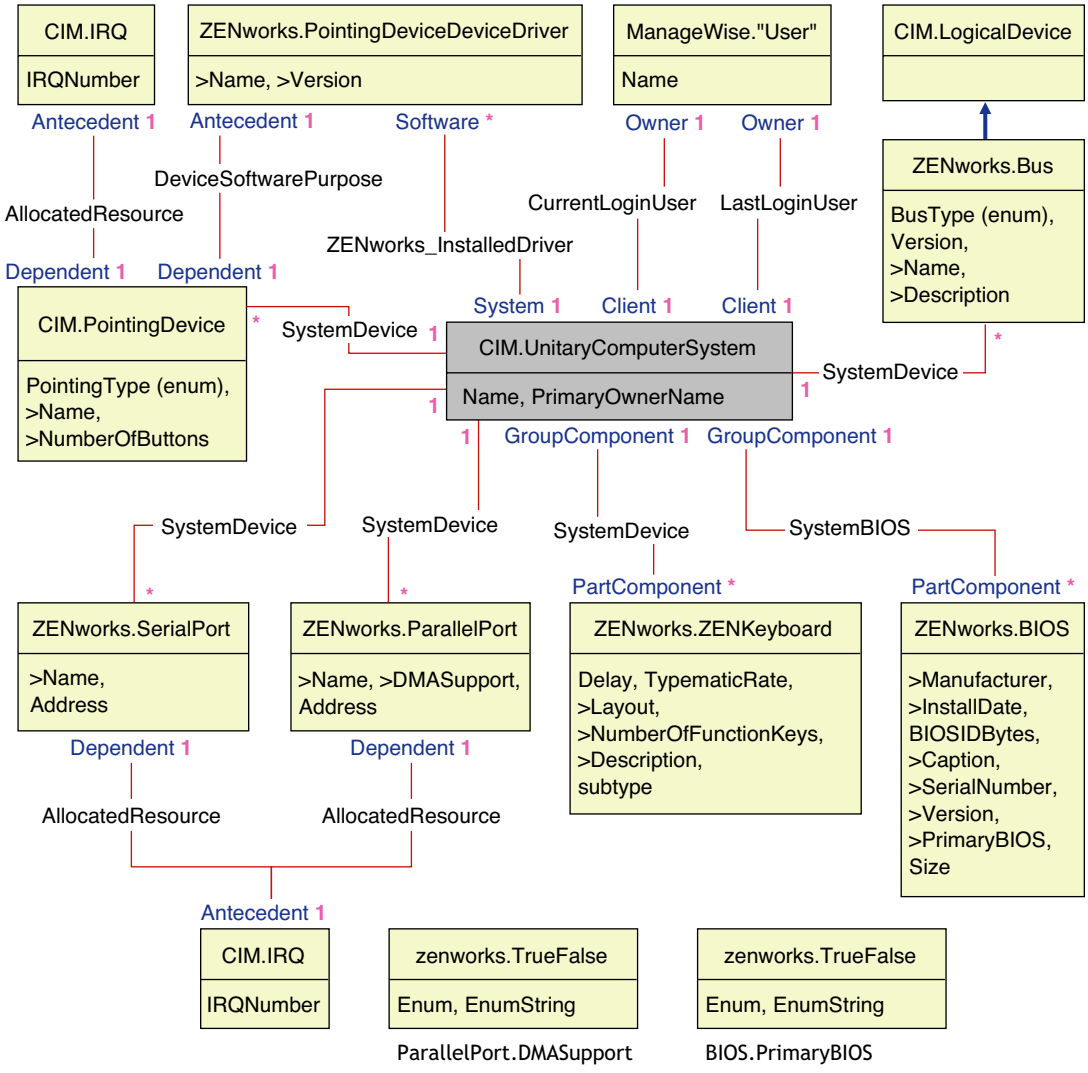
Schema Diagrams of CIM and the Extension Schema in ZENworks 6.5 Server Management

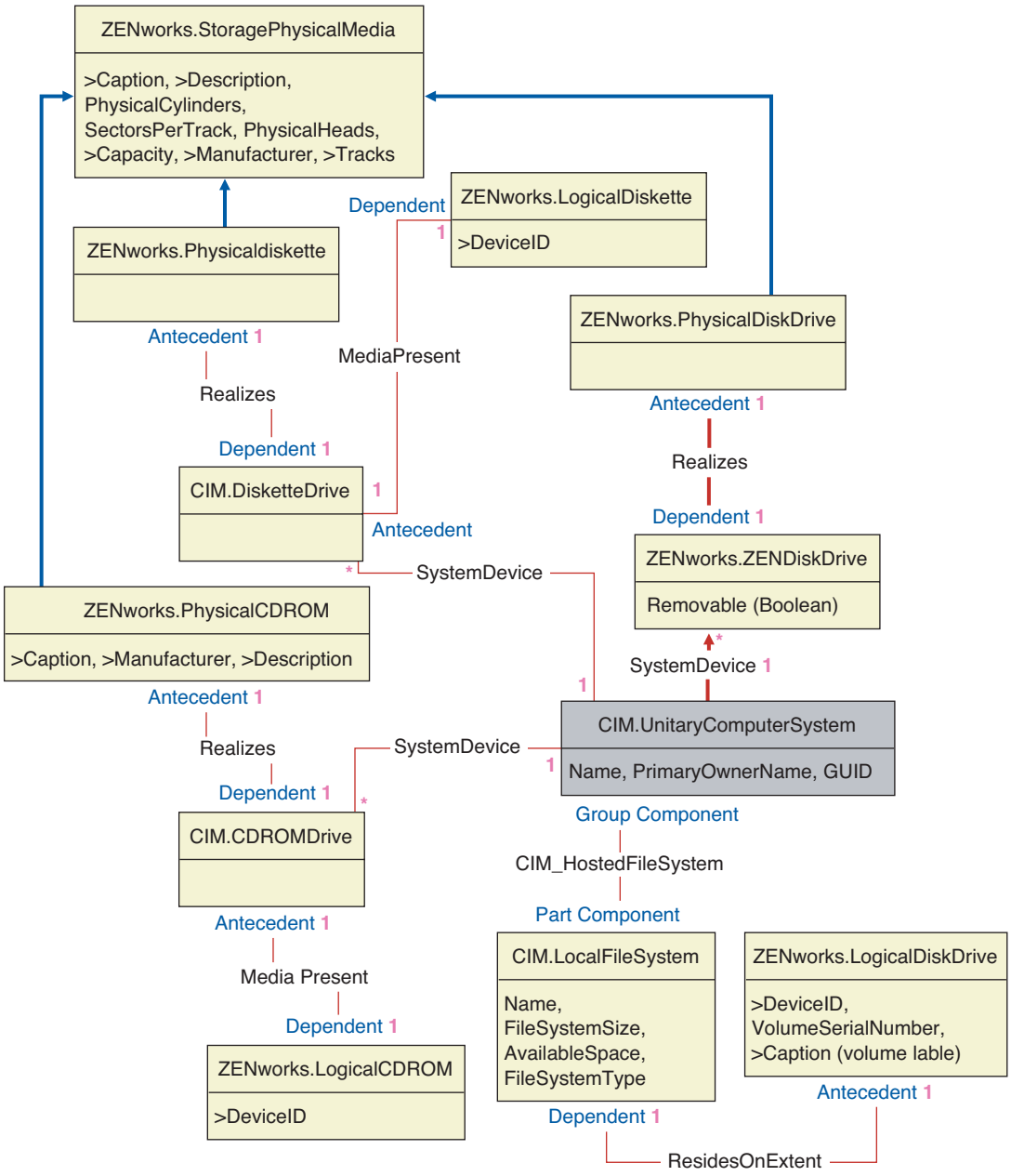
The following schema diagrams of the CIM and extension schema model the Inventory database in ZENworks 6.5 Server Management.

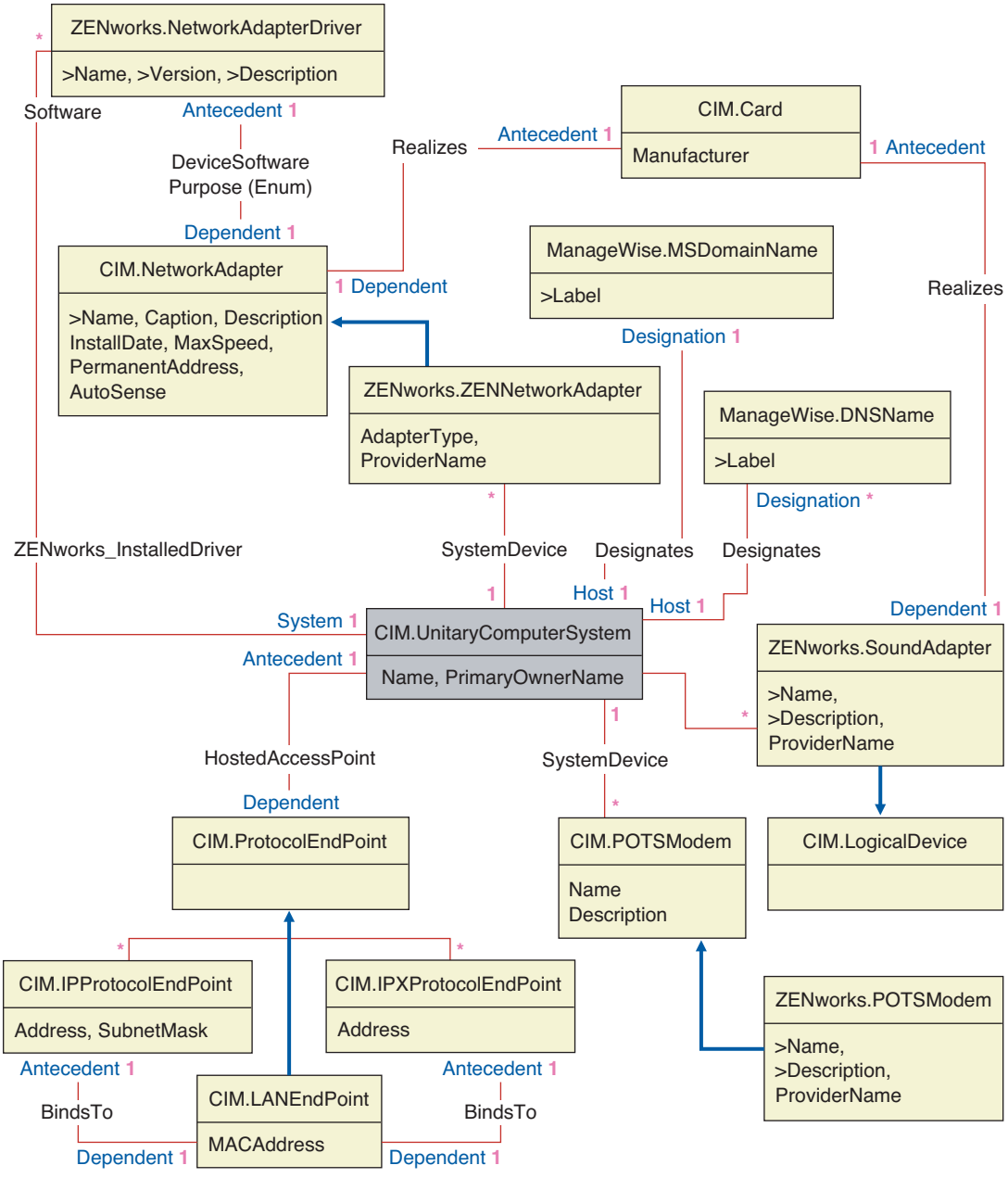

CIM.Processor
Stepping, DeviceID, Family (Enum), OtherFamilyDescription, MaxClockSpeed, CurrentClockSpeed, Role (Enum), UpgradeMethod (Enum)
CIM.VideoBIOSElement
Manufacturer, Version, InstallDate, isShadowed
1
Software *
PartComponent *
ZENworks_InstalledDriver
ComputerSystemProcessor
GroupComponent 1
System 1
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName
DeviceSoftware Purpose
1
GroupComponent 1
SystemDevice
InstalledOS
1
*
PartComponent
ZENworks.VideoAdaptor
>Description,
>NumberOfColorPlanes,
>CurrentHorizontalResolution,
>CurrentVerticalResolution,
>VideoArchitecture,
>VideoMemoryType,
>MaxMemorySupported,
>CurrentBitsPerPixel,
>MaxRefreshRate,
>MinRefreshRate,
DACType
ChipSet
ProviderName
ZENworks.ZENOperatingSystem
>OSType (Role),
>Caption,
CodePage,
Role (Enum),
>Version,
>InstallDate,
>TotalVirtualMemorySize,
>TotalVisibleMemorySize,
ProviderName
1
Realizes
ZENworks.WinOperatingSystem
CIM.Card
Manufacturer
Other Operating Systems
ZENworks.NetWareOperatingSystem
AccountingVersion,
InternetBridgeSupport,
MaxNumberOfVolumes,
PeakConnectionsUsed,
PrintServerVersion,
QueuingVersion,
RevisionLevel,
SecurityRestrictionLevel,
SFTLevel,
TTSLevel,
VAPVersion,
VirtualConsoleVersion,
InternalNetworkNumber
ZENworks.UnixOperatingSystem
KernelVersion

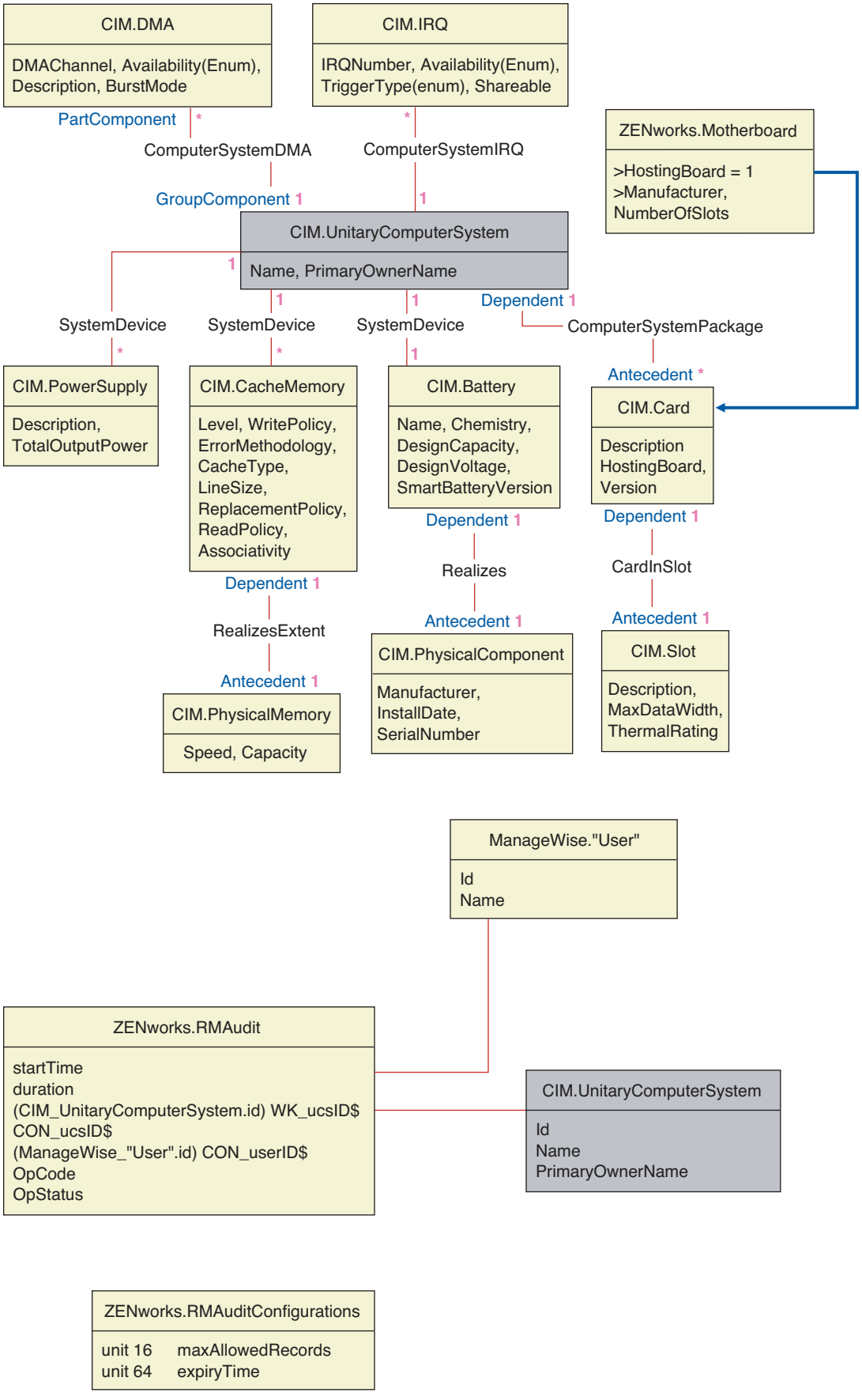

ZENworks.inventoryScanner
>Version, LastScanDate,
InventoryServer, ScanMode (Role),
GeneralDictionaryVersion,
PrivateDictionaryVersion
Software 1
CMI_InstalledSoftwareElement
System 1
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName
ZENworks.ZENChassis
AssetTag
ChassisType (Enum CMI_ChassisType)
>NumberOfPowerCords (CIM_Chassis)
>Manufacturer (CIM_PhysicalElement)
>SerialNumber (CIM_PhysicalElement)
>Version (CIM_PhysicalElement)
>Tag (CIM_PhysicalElement)
Antecedent *
ComputerSystemPackage
Dependent *
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName
Host 1
Dependent 1
Designates
ComputerSystemPackage
Designation 1
Antecedent 1
ManageWise.NDSName
>Label
Tree
ZENworks.SystemInfo
> Tag
AssetTag
ModelNumber
>Model,
>SerialNumber,
>Manufacturer,
>Description,
SystemIdentifer,
ManagementTechnology(Enum)


CIM.PhysicalComponent
SerialNumber
Manufacturer
Model
Antecedent *
CIM_Realizes
Dependent *
ZENworks.ZENDesktopMonitor
>DeviceID(CMI_LogicalDevice)
>Description (CIM_ManagedSystemElement)
ModleIID
ManufacturerDate
ViewableSize
NominalSize
PartComponent *
CIM_SystemDevice
GroupComponent 1
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName


CIM.IRQ
IRQNumber
ZENworks.PointingDeviceDeviceDriver
>Name, >Version
ManageWise."User"
Name
CIM.LogicalDevice
ZENworks.Bus
BusType (enum),
Version,
>Name,
>Description
Antecedent 1
Antecedent 1
Software *
Owner 1
Owner 1
DeviceSoftwarePurpose
AllocatedResource
CurrentLoginUser
LastLoginUser
ZENworks_InstalledDriver
Dependent 1
Dependent 1
System 1
Client 1
Client 1
CIM.PointingDevice
PointingType (enum),
>Name,
>NumberOfButtons
SystemDevice
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName
SystemDevice
GroupComponent 1
GroupComponent 1
SystemDevice
SystemDevice
SystemDevice
SystemBIOS
PartComponent *
PartComponent *
ZENworks.SerialPort
>Name,
Address
ZENworks.ParallelPort
>Name, >DMASupport,
Address
ZENworks.ZENKeyboard
Delay, TypematicRate,
>Layout,
>NumberOfFunctionKeys,
>Description,
subtype
ZENworks.BIOS
>Manufacturer,
>InstallDate,
BIOSIDBytes,
>Caption,
>SerialNumber,
>Version,
>PrimaryBIOS,
Size
Dependent 1
Dependent 1
AllocatedResource
AllocatedResource
Antecedent 1
CIM.IRQ
IRQNumber
zenworks.TrueFalse
Enum, EnumString
ParallelPort.DMASupport
zenworks.TrueFalse
Enum, EnumString
BIOS.PrimaryBIOS


ZENworks.StoragePhysicalMedia
>Caption, >Description,
PhysicalCylinders,
SectorsPerTrack, PhysicalHeads,
>Capacity, >Manufacturer, >Tracks
ZENworks.Physicaldiskette
Antecedent 1
Realizes
Dependent 1
Dependent
1
ZENworks.LogicalDiskette
>DeviceID
MediaPresent
ZENworks.PhysicalDiskDrive
Antecedent 1
Realizes
Dependent 1
CIM.DisketteDrive
1
Antecedent
*
SystemDevice
ZENworks.ZENDiskDrive
Removable (Boolean)
*
SystemDevice 1
1
ZENworks.PhysicalCDROM
>Caption, >Manufacturer, >Description
Antecedent 1
Realizes
Dependent 1
SystemDevice
1
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName, GUID
Group Component
CIM_HostedFileSystem
Part Component
*
CIM.CDROMDrive
Antecedent 1
Media Present
Dependent 1
ZENworks.LogicalCDROM
>DeviceID
CIM.LocalFileSystem
Name,
FileSystemSize,
AvailableSpace,
FileSystemType
Dependent 1
ZENworks.LogicalDiskDrive
>DeviceID,
VolumeSerialNumber,
>Caption (volume lable)
Antecedent 1
ResidesOnExtent

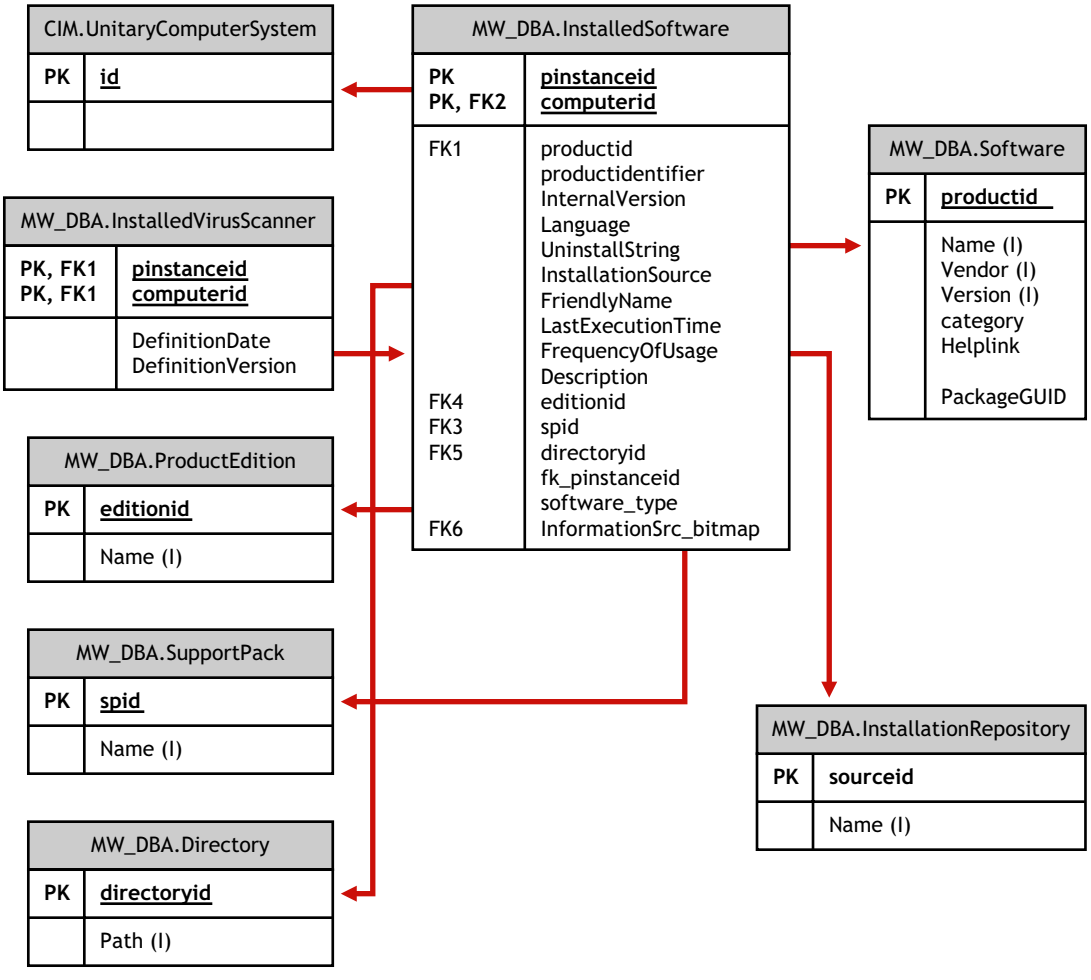

ZENworks.NetworkAdapterDriver
>Name, >Version, >Description
Software
Antecedent 1
DeviceSoftware
Purpose (Enum)
Dependent 1
CIM.NetworkAdapter
>Name, Caption, Description
InstallDate, MaxSpeed,
PermanentAddress,
AutoSense
Realizes
Antecedent 1
CIM.Card
Manufacturer
1 Antecedent
Realizes
1 Dependent
ManageWise.MSDomainName
>Label
Designation 1
ZENworks.ZENNetworkAdapter
AdapterType,
ProviderName
ManageWise.DNSName
>Label
Designation *
ZENworks_InstalledDriver
SystemDevice
Designates
Designates
System 1
Host 1
Host 1
Dependent 1
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName
Antecedent 1
ZENworks.SoundAdapter
>Name,
>Description,
ProviderName
HostedAccessPoint
SystemDevice
Dependent
CIM.ProtocolEndPoint
CIM.POTSModem
Name
Description
CIM.LogicalDevice
CIM.IPProtocolEndPoint
Address, SubnetMask
CIM.IPXProtocolEndPoint
Address
ZENworks.POTSModem
>Name,
>Description,
ProviderName
Antecedent 1
Antecedent 1
BindsTo
BindsTo
CIM.LANEndPoint
MACAddress
Dependent 1
Dependent 1


CIM.DMA
DMAChannel, Availability(Enum), Description, BurstMode
CIM.IRQ
IRQNumber, Availability(Enum), TriggerType(enum), Shareable
PartComponent
*
*
ComputerSystemDMA
ComputerSystemIRQ
ZENworks.Motherboard
>HostingBoard = 1
>Manufacturer, NumberOfSlots
GroupComponent 1
1
CIM.UnitaryComputerSystem
Name, PrimaryOwnerName
1
1
1
Dependent 1
SystemDevice
SystemDevice
SystemDevice
ComputerSystemPackage
*
*
1
Antecedent *
CIM.PowerSupply
Description, TotalOutputPower
CIM.CacheMemory
Level, WritePolicy, ErrorMethodology, CacheType, LineSize, ReplacementPolicy, ReadPolicy, Associativity
CIM.Battery
Name, Chemistry, DesignCapacity, DesignVoltage, SmartBatteryVersion
CIM.Card
Description HostingBoard, Version
Dependent 1
Dependent 1
Realizes
CardInSlot
Dependent 1
Antecedent 1
Antecedent 1
RealizesExtent
CIM.PhysicalComponent
Manufacturer, InstallDate, SerialNumber
CIM.Slot
Description, MaxDataWidth, ThermalRating
Antecedent 1
CIM.PhysicalMemory
Speed, Capacity
ManageWise."User"
Id
Name
ZENworks.RMAudit
startTime
duration
(CIM_UnitaryComputerSystem.id) WK_ucsID$
CON_ucsID$
(ManageWise_"User".id) CON_userID$
OpCode
OpStatus
CIM.UnitaryComputerSystem
Id
Name
PrimaryOwnerName
ZENworks.RMAuditConfigurations
unit 16 maxAllowedRecords
unit 64 expiryTime

Software Inventory Schema

The following software schema diagrams model the Inventory database in ZENworks 6.5 Server Management. In the following schema diagram, the CIM.UnitaryComputerSystem represents a managed inventory system.

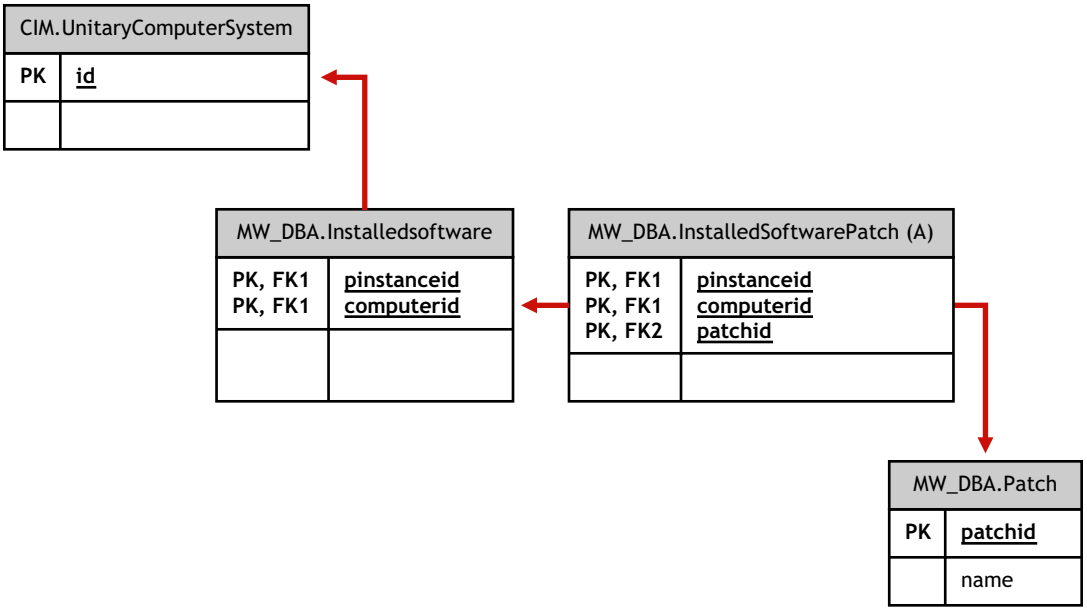
For more information about the tables, see Appendix J, “ZENworks 6.5 Server Management Inventory Attributes,” on page 705.



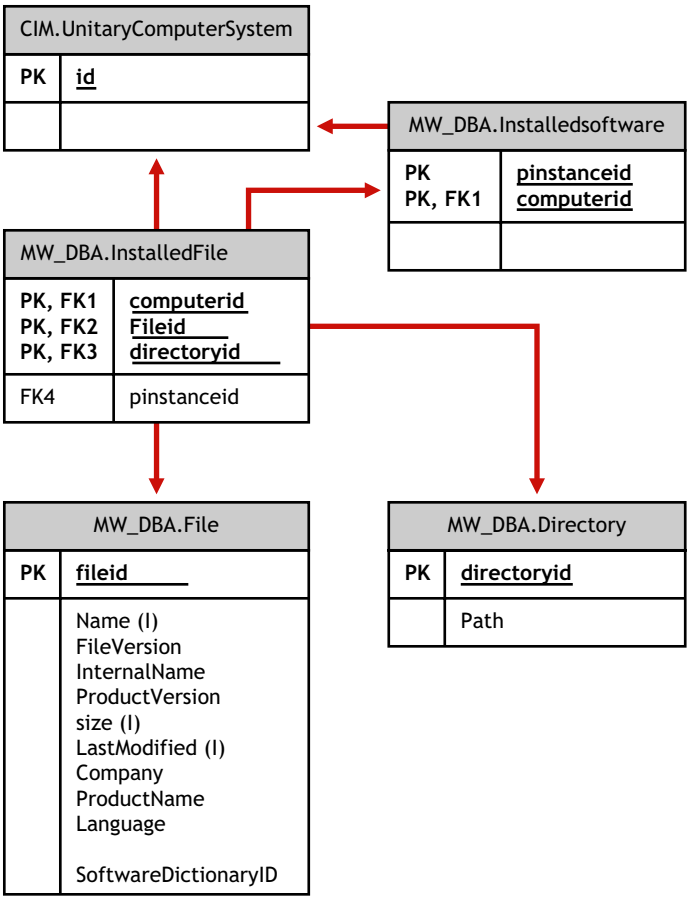
In the above illustration, class MW_DBA.Software associates to CIM.UnitaryComputerSystem using the association MW_DBA.InstalledSoftware with MW_DBA.InstalledSoftware.ComputerSystem pointing to CIM.UnitaryComputerSystem and MW_DBA.InstalledSoftware.ProductID pointing to MW_DBA.Software. The relationship between the two classes is one to many. This means a computer system might have more than one software information.

MW_DBA.InstalledSoftware association has Foreign key references to the following tables: ProductEdition, SupportPack, Directory, and Installation Repository.

MW_DBA.InstalledVirusScanner inherits the software information from MW_DBA.InstalledSoftware along with virus specific information such as Definition date and Definition version.

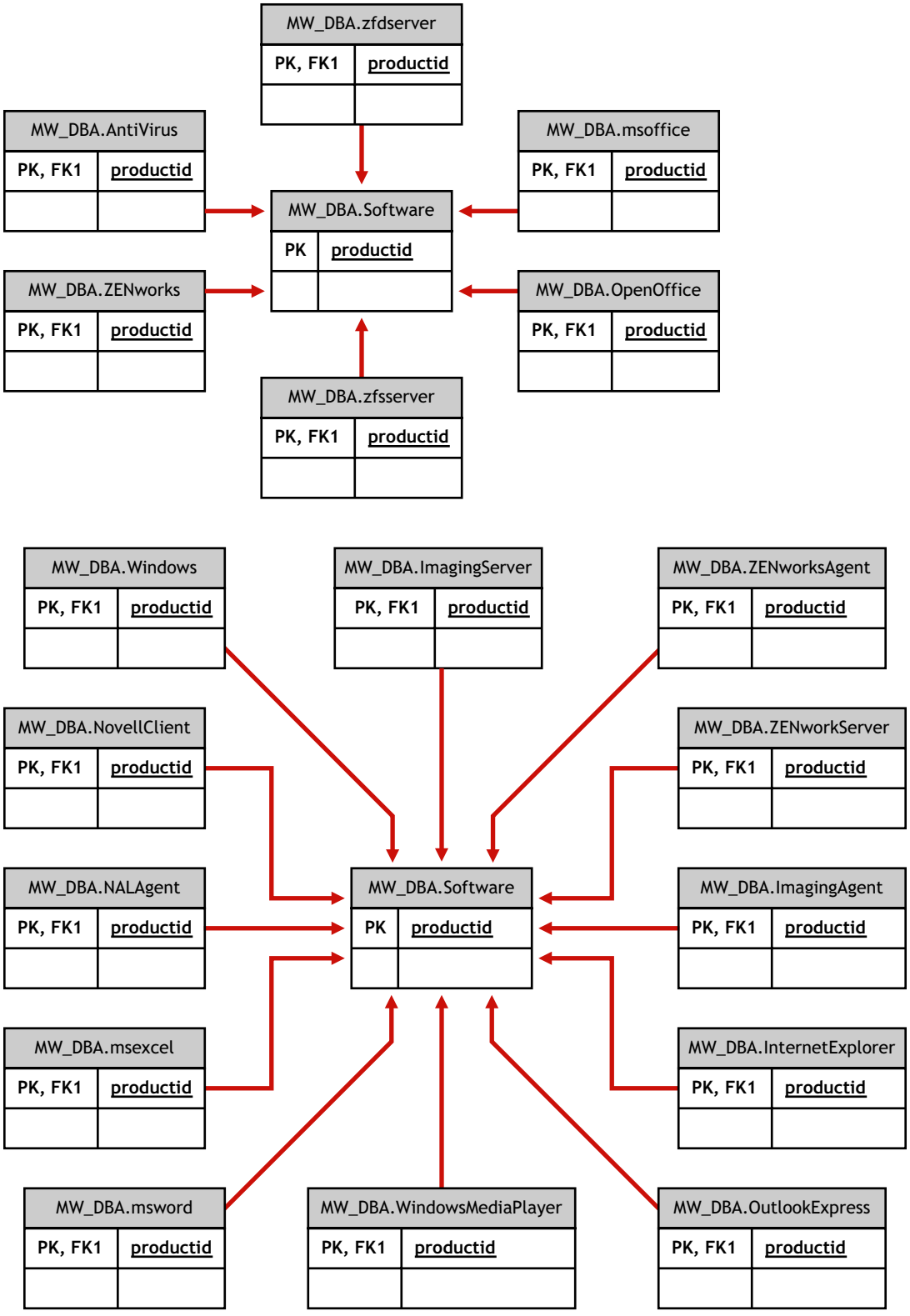


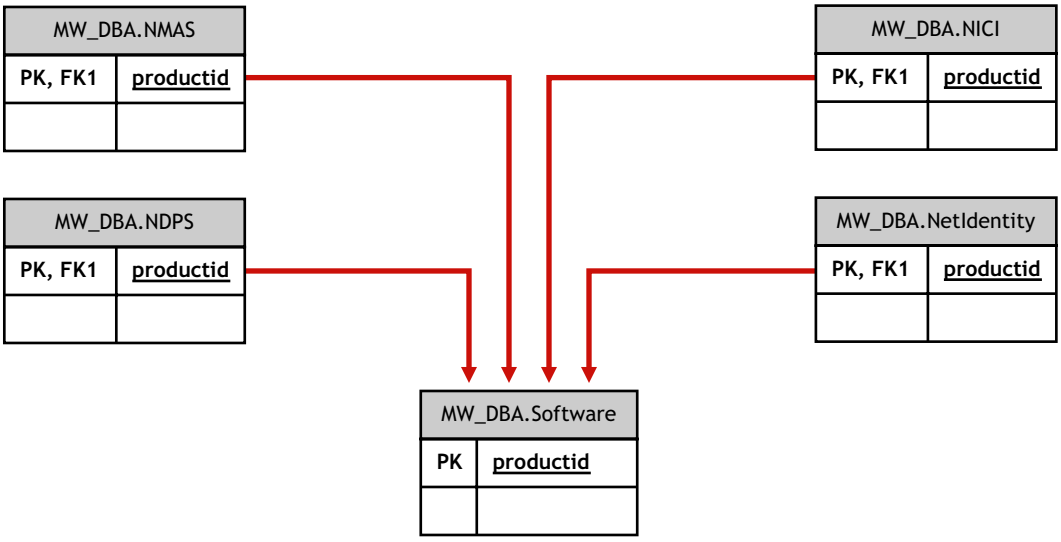
In the above illustration, class MW_DBA.Patch associates to MW_DBA.InstalledSoftware using the association MW_DBA.InstalledSoftwarePatch with MW_DBA.InstalledSoftwarePatch.pinstanceID pointing to MW_DBA.InstalledSoftware and MW_DBA.InstalledSoftwarePatch.PatchID pointing to MW_DBA.Patch. The relationship between the two classes is one to many. This means a software might have zero or more patch information.



In the above illustration, class MW_DBA.File associates to MW_DBA.InstalledSoftware using the association MW_DBA.InstalledFile with MW_DBA.InstalledFile.pinstanceID pointing to MW_DBA.InstalledSoftware and MW_DBA.InstalledFile.fileID pointing to MW_DBA.File. The relationship between the two classes is one to many. This means a software might have zero or more file information.

In this illustration, class MW_DBA.Directory associates to MW_DBA.InstalledSoftware using the association MW_DBA.InstalledFile with MW_DBA.InstalledFile.pinstanceID pointing to MW_DBA.InstalledSoftware and MW_DBA.InstalledFile.DirectoryID pointing to MW_DBA.Directory.

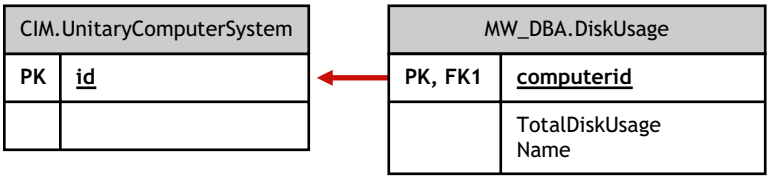

MW_DBA.zfdserver
PK, FK1
productid
MW_DBA.AntiVirus
PK, FK1
productid
MW_DBA.msoffice
PK, FK1
productid
MW_DBA.Software
PK
productid
MW_DBA.ZENworks
PK, FK1
productid
MW_DBA.OpenOffice
PK, FK1
productid
MW_DBA.zfsserver
PK, FK1
productid
MW_DBA.Windows
PK, FK1
productid
MW_DBA.ImagingServer
PK, FK1
productid
MW_DBA.ZENworksAgent
PK, FK1
productid
MW_DBA.NovellClient
PK, FK1
productid
MW_DBA.ZENworkServer
PK, FK1
productid
MW_DBA.NALAgent
PK, FK1
productid
MW_DBA.Software
PK
productid
MW_DBA.ImagingAgent
PK, FK1
productid
MW_DBA.msexcel
PK, FK1
productid
MW_DBA.InternetExplorer
PK, FK1
productid
MW_DBA.msword
PK, FK1
productid
MW_DBA.WindowsMediaPlayer
PK, FK1
productid
MW_DBA.OutlookExpress
PK, FK1
productid



In the above illustrations, MW_DBA.MSoffice inherits the software information from MW_DBA.SOftware. This sub-class gets directly the MS office information.

This is also applicable for the following classes:

mw_dba.zfdserver
mw_dba.zfsserver
mw_dba.zfsinventoryserver
mw_dba.zfsinventoryagent
mw_dba.zfsrmserver
mw_dba.zfsrmagent
mw_dba.zfdrmserver
mw_dba.zfdrmagent
mw_dba.zfsinvxmlproxyserver
mw_dba.zfdnalagent
mw_dba.middletier
mw_dba.zfspxeserver
mw_dba.zfsmmsadctrendingagent
mw_dba.zfhaccesspoin
mw_dba.zfdinventoryserver
mw_dba.zfdinventoryagent
mw_dba.mspowerpoint
mw_dba.msoutlook
mw_dba.msaccess
mw_dba.mspublisher
mw_dba.msfrontpage
mw_dba.msinfopath
mw_dba.zfdimagingagent
mw_dba.zfdnalserver
mw_dba.zfsmmsserver
mw_dba.zfsmmssrvmgmtagent
mw_dba.zfspdsdb
mw_dba.zfhdesktopsync
mw_dba.zfdagent
mw_dba.zfsagent
mw_dba.msphotodraw
mw_dba.zfdwsmanager
mw_dba.zfdwsimportserver
mw_dba.zfdinvdbserver
mw_dba.zfsinvdbserver
mw_dba.zfdinvxmlproxyserver
mw_dba.zfdimagingserver
mw_dba.zfdnaldb
mw_dba.zfspds
mw_dba.zfsmmstrafficanalysisagent
mw_dba.zfhserver



In the above illustration, MW_DBA.DiskUsage has the computerID column foreign key references to the CIM.UnitaryComputerSystem.ID. The MW_DBA.DiskUsage table contains the total disk usage and the file extension name.

Sample Inventory Database Queries

The following are sample queries for retrieving the inventory information from the ZENworks 6.5 Server Management Inventory database.

Refer to the schema diagrams in “Schema Diagrams of CIM and the Extension Schema in ZENworks 6.5 Server Management” on page 523 to find out the associated schema classes and attribute information.

1. Retrieve the name and ID of all inventoried servers from the database and also to the eDirectory tree to which these servers are registered. The query is as follows:

```
SELECT
  u.id$, u.name, m.tree
FROM
  ManageWise.NDSName m,
  CIM.UnitaryComputerSystem u,
  ManageWise.Designates s
WHERE
  s.Designation=m.id$ AND s.Host=u.id$;
```

   In the above query, the tree name is part of the computer system name.

2. Retrieve the asset tag, manufacturer, and model number of all the inventoried servers in the database. The query is as follows:

```
SELECT
  m.AssetTag,
  m.Manufacturer,
  m.ModelNumber,
  m.SerialNumber
FROM
  CIM.UnitaryComputerSystem u,
  CIM.ComputerSystemPackage s,
  ZENworks.SystemInfo m
WHERE
  s.Antecedent=m.id$ AND s.Dependent=u.id$;
```

3. Retrieve all the Microsoft applications with their versions and IDs that are installed on the inventoried server 'SJOHN164_99_139_79' registered under the 'NOVELL_AUS' eDirectory tree. The query is as follows:

```
SELECT
  m.Name,
  m.Version,
  im.ProductIdentifier
```

```
FROM
  CIM.UnitaryComputerSystem u,
  MW_DBA.InstalledSoftware im,
  MW_DBA.Software m
WHERE
  u.Name='SJOHN164_99_139_79.Novell_AUS' AND
  (im.computerid=u.id$ and im.productid=m.productid)
  AND m.Vendor LIKE 'Microsoft%';
```

4. Retrieve the processor information for the inventoried server 'SJOHN164_99_139_79.NOVELL_AUS'. The query is as follows:

```
SELECT
  procr.DeviceID,
  role.EnumString,
  family.EnumString,
  procr.OtherFamilyDescription,
  upg.EnumString,
  procr.MaxClockSpeed,
  procr.CurrentClockSpeed,
  procr.Stepping
FROM
  CIM.UnitaryComputerSystem ucs,
  CIM.ComputerSystemProcessor csp,
  CIM.Processor procr,
  CIM.Role_en_US role,
  CIM.Family_en_US family,
  CIM.UpgradeMethod_en_US upg
WHERE
 ucs.name='SJOHN164_99_139_79.Novell_AUS' AND
  csp.PartComponent=procr.id$ AND
(
  (
    ( procr.Role IS NOT NULL AND procr.Role=role.Enum ) OR
    ( procr.Role IS NULL AND role.Enum=1000 )
  )
  AND
```

```
procr.Family=family.Enum
  AND
  (
   ( procr.UpgradeMethod IS NOT NULL AND procr.UpgradeMethod=upg.Enum ) OR
    ( procr.UpgradeMethod IS NULL AND upg.Enum=1000 )
  )
);
```

5. Retrieve the ID of the UnitaryComputerSystem used for the inventoried server 'SJOHN164_99_139_79.NOVELL_AUS'. The query is as follows:

```
SELECT
  id$
FROM
  CIM.UnitaryComputerSystem
WHERE
  Name='SJOHN164_99_139_79.Novell_AUS';
```

6. Find the number of inventoried servers in the database. The query is as follows:

```
SELECT
  count(u.id$)
FROM
  CIM.UnitaryComputerSystem u,
  CIM.InstalledSoftwareElement s,
  ZENworks.InventoryScanner m
WHERE
  m.id$=s.Software AND u.id$=s.System;
```

7. When you know the ID of the UnitaryComputerSystem for a particular inventoried server from the query as shown in query 5, query 4 can be modified as:

```
SELECT
procr.DeviceID,
role.EnumString,
family.EnumString,
procr.OtherFamilyDescription,
upg.EnumString,
procr.MaxClockSpeed,
procr.CurrentClockSpeed,
procr.Stepping
FROM
```

```
CIM.UnitaryComputerSystem ucs,
CIM.ComputerSystemProcessor csp,
CIM.Processor procr,
CIM.Role_en_US role,
CIM.Family_en_US family,
CIM.UpgradeMethod_en_US upg
WHERE
ucs.id$ = ? AND
csp.PartComponent=procr.id$ AND
(
  (
    ( procr.Role IS NOT NULL AND procr.Role=role.Enum ) OR
    ( procr.Role IS NULL AND role.Enum=1000 )
  )
  AND
procr.Family=family.Enum
  AND
  (
   ( procr.UpgradeMethod IS NOT NULL AND procr.UpgradeMethod=upg.Enum ) OR
    ( procr.UpgradeMethod IS NULL AND upg.Enum=1000 )
  )
);
```

Substitute the ID of the specified inventoried server in place of the ?, value for ucs.id$ in the query.

8. List the IP address, IPX address, and MAC address of all servers in the database. The query is as follows:

```
SELECT
  u.name,
  ip.Address,
  ipx.Address,
  mac.MACAddress
FROM
  CIM.UnitaryComputerSystem u,
  CIM.HostedAccessPoint s1,
  CIM.IPProtocolEndpoint ip,
  CIM.HostedAccessPoint s2,
```

```
  CIM.IPXProtocolEndpoint ipx,
  CIM.HostedAccessPoint s3,
  CIM.LANEndpoint mac
WHERE
  (s1.Dependent=ip.id$ and s1.Antecedent=u.id$) AND
  (s2.Dependent=ipx.id$ and s2.Antecedent=u.id$) AND
  (s3.Dependent=mac.id$ and s3.Antecedent=u.id$);
```

9. Retrieve the name and other properties of the drives on the hard disk of the specified inventoried server. The query is as follows:

```
SELECT
  n.Name,
  m.DeviceID,
  n.FileSystemSize,
  n.AvailableSpace,
  n.FileSystemType,
  m.VolumeSerialNumber,
  m.caption as VolumeLabel
FROM
  CIM.HostedFileSystem s,
  CIM.LocalFileSystem n,
  CIM.ResidesOnExtent r,
  ZENworks.LogicalDiskDrive m
WHERE
  (s.GroupComponent=? and s.PartComponent=n.id$) AND
  (r.Dependent=n.id$ and r.Antecedent=m.id$);
```

10. Retrieve all Custom attribute information stored in the database. The query is as follows:

```
SELECT * FROM ZENworks.CustomInformation;
```

11. Retrieve all Custom attribute information associated to the Class CIM.UnitaryComputerSystem. The query is as follows:

```
SELECT
  *
FROM
  ZENworks.CustomInformation
WHERE
  extractClass(id) IN
```

```
(SELECT id FROM MW_DBA.t$Class WHERE
ClassName='CIM. UnitaryComputerSystem')
```

12. Retrieve all the Microsoft Office installations in the enterprise. The query is as follows:

```
SELECT

  u.name,

  m.FriendlyName,

  im.InternalVersion,

  im.ProductIdentifier

FROM

  CIM.UnitaryComputerSystem u,

  MW_DBA.InstalledSoftware im,

  MW_DBA.Software m,

  MW_DBA.MSOffice mso

WHERE

  mso.id$=m.productid AND

  m.productid=im.productid AND

  im.computerid=u.id$;
```

13. Retrieve all the Internet Explorer installations in the enterprise.. The query is as follows:

```
SELECT

  u.Name,

  m.Name,

  m.Version,

  im.InternalVersion,

  im.ProductIdentifier

FROM

  CIM.UnitaryComputerSystem u,

  MW_DBA.InstalledSoftware im,

  MW_DBA.Software m,

  MW_DBA.InternetExplorer ie

WHERE

  ie.id$=m.productid AND

  m.productid=im.productid AND

  im.computerid=u.id$;
```

NOTE: Query 12 and 13 follow nearly the same syntax except for the table relating to the component. A similar approach can be used for the following components as well, Windows Media Player, Outlook Express, Microsoft Word, Microsoft Excel etc. The complete set of these tables are available in the Schema.

14. Retrieve all the Anti-Virus installations in the enterprise. The query is as follows:

```
SELECT
  u.Name,
  m.Name,
  m.Version,
  im.InternalVersion,
  ivs.DefinitionVersion,
  ivs.DefinitionDate
FROM
  CIM.UnitaryComputerSystem u,
  MW_DBA.InstalledSoftware im,
  MW_DBA.Software m,
  MW_DBA.InstalledVirusScanner ivs
WHERE
  ivs.pinstanceid=im.pinstanceid AND
  m.productid=im.productid AND
  im.computerid=u.id$;
```

15. Retrieve all the applications and the details of the files associated with the application that are installed on the inventoried server 'SJOHN164_99_139_79.NOVELL_AUS'. The query is as follows:

```
SELECT
  u.Name,
  m.Name,
  m.Version,
  m.Category,
  zfile.company,
  zfile.productname,
  zfile.productversion,
  zfile.name,
  dir.path,
  zfile.fileversion,
  zfile."size",
  zfile.lastmodified,
  zfile.internalname,
  zfile.softwaredictionaryid
FROM
```

```
  CIM.UnitaryComputerSystem u,
  MW_DBA.InstalledSoftware iso,
  MW_DBA.Software m,
  MW_DBA.InstalledFile ifile,
  MW_DBA."file" zfile,
  MW_DBA.Directory dir
WHERE
  u.Name='SJOHN164_99_139_79.Novell_AUS' AND
  iso.computerid=u.id$ AND
  iso.productid=m.productid AND
  iso.pinstanceid=ifile.pinstanceid AND
  ifile.directoryid=dir.id AND
  ifile.fileid=zfile.id;
```

16. Retrieve all the files present on the inventoried server 'SJOHN164_99_139_79.NOVELL_AUS' which has not been associated with a valid software. The query is as follows:

```
SELECT
  u.Name,
  zfile.name,
  dir.path,
  zfile.fileversion,
  zfile."size",
  zfile.lastmodified,
  zfile.internalname,
  zfile.productversion,
  zfile.company,
  zfile.productname
FROM
  CIM.UnitaryComputerSystem u,
  MW_DBA.InstalledFile ifile,
  MW_DBA."file" zfile,
  MW_DBA.Directory dir
WHERE
  u.Name='SJOHN164_99_139_79.Novell_AUS' AND
  u.id$=ifile.computerid AND
  ifile.fileid=zfile.id AND
```

```
  ifile.directoryid=dir.id AND

  ifile.pinstanceid is null;
```

17. Retrieve the disk usage details of files with known extensions on each inventories machine in the enterprise. The query is as follows:

```
SELECT

  u.Name,

  du.Name,

  du.TotalDiskUsage

FROM

  CIM.UnitaryComputerSystem u,

  MW_DBA.DiskUsage du

WHERE

  u.id$=du. Computerid AND

  du.Name is not null;
```

16 Managing Your Inventory Information

This section contains the following information to help you customize the way Novell® ZENworks® 6.5 Server Inventory displays information:

Viewing the Inventory Servers Deployed for Inventory

Using ConsoleOne®, you can view the Inventory servers and databases that you configured for collecting inventory.

To get a complete Inventory tree view, you need to log into all the Novell eDirectory™ trees that contain Inventory servers present in your inventory tree. In ConsoleOne, select a container, click the View menu, then click Complete Tree View. All the Inventory servers within the container are displayed in the Complete Tree View.

To view a complete tree view if your inventory deployment involves roll-up of information between Inventory servers that are situated on different Novell eDirectory trees:

1. In ConsoleOne, select NDS Tree.
2. Click View, then click Complete Tree View.
3. Select the eDirectory trees or containers within the tree that contains the Inventory servers.
4. Click OK.

To view all Inventory server from the selected Inventory server to the highest-level server, in ConsoleOne, right-click the Inventory Service object (Inventory Service_*server_name*), click View, then click Up Tree View or double-click the Inventory Service object.

If your inventory deployment consists of a single eDirectory tree, an Up Tree View displays all the Inventory servers from the selected Inventory server up to the highest level (Root Server).

If your inventory deployment involves roll-up of inventory information across Inventory servers located on different eDirectory trees, the Up Tree View displays all the Inventory servers from the selected Inventory server up to the highest level server to which you have logged in.

NOTE: You cannot collapse the inventory tree using the short-cut keys.

Customizing the Hardware Inventory Information To Be Scanned

ZENworks 6.5 Server Management allows you to collect information that is not part of the default hardware inventory from the inventoried servers.

Scanning for Vendor-Specific Asset Information from DMI

1. In the Server Inventory policy. click the Configuration Editor tab.

   For more information, see "Configuring the Server Inventory Policy" on page 486.

2. Click the Asset Information suboption, then click Set Defaults.

   The following entries will be populated.

   ```
   [ASSETTAG]
   DMI1_CLASSNAME=
   DMI1_ATTRIBUTEID=
   DMI2_CLASSNAME=
   DMI2_ATTRIBUTEID=
   [SERIALNUMBER]
   DMI1_CLASSNAME=
   DMI1_ATTRIBUTEID=
   DMI2_CLASSNAME=
   DMI2_ATTRIBUTEID=
   [MODEL]
   DMI1_CLASSNAME=
   DMI1_ATTRIBUTEID=
   DMI2_CLASSNAME=
   DMI2_ATTRIBUTEID=
   [COMPUTERTYPE]DMI1_CLASSNAME=DMI1_ATTRIBUTEID=
   [MODELNUMBER]DMI1_CLASSNAME=DMI1_ATTRIBUTEID=
   ```

3. Specify the values.

   The Asset Information contains the following sections:

   - Contains Asset Tag in the section [ASSETTAG]
   - Contains Serial Number in the section [SERIALNUMBER]
   - Contains Computer Model in the section [MODEL]
   - Contains Computer Type [COMPUTERTYPE]
   - Contains Computer Model Number [MODELNUMBER]

Each section contains the particular DMI Class name and DMI Class Attribute ID.

The format of Asset Information is as follows:

```
[ASSETTAG]
DMI1_CLASSNAME=DMI_class_name_for_asset_tag
DMI1_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag
[SERIALNUMBER]
DMI1_CLASSNAME=DMI_class_name_for_serial_number
DMI1_ATTRIBUTEID=DMI_attribute_ID_for_serial_number
[MODEL]
DMI1_CLASSNAME=DMI_class_name_for_computer_model
DMI1_ATTRIBUTEID=DMI_attribute_ID_for_computer_model
```

The value of the Asset Information sections can have a maximum string length of 64 characters.

A DMI Class name can be any DMI class other than DMTF|COMPONENTID|00x.

If there is more than one DMI vendor implementing different custom DMI classes, you can specify multiple DMI classes. A maximum of five classes can be specified in these sections. For example, the asset information for five classes is as follows:

```
[ASSETTAG]
DMI1_CLASSNAME=DMI_class_name_for_asset_tag
DMI1_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag
DMI2_CLASSNAME=DMI_class_name_for_asset_tag
DMI2_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag
DMI3_CLASSNAME=DMI_class_name_for_asset_tag
DMI3_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag
DMI4_CLASSNAME=DMI_class_name_for_asset_tag
DMI4_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag
DMI5_CLASSNAME=DMI_class_name_for_asset_tag
DMI5_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag
```

The scanner will processes DMI1 and if the values of DMI1 are valid, the scanner will not process the remaining DMI classes.

4. Click OK.
5. Run the scans on the inventoried servers.

   Verify that the inventory information is in the Inventory Summary window.

Customizing the Hardware Scanning Information of Jaz and Zip Drive Vendors

The scan information of the vendors for devices such as backup and floppy devices is usually unavailable on the inventoried server. Also, if the information is available, the vendor information does not usually contain the details. You can customize and update information about the vendors of these devices in Server Inventory policy > Configuration Editor > Zipped Names. The scanners read this information during the hardware scanning process for these devices.

To customize and update the vendor information for display:

1. In the Server Inventory policy, click the Configuration Editor tab.

   For more information, see “Configuring the Server Inventory Policy” on page 486.

2. Click the ZIPPED NAMES suboption, then click Set Defaults.

   The default values are displayed.

   ```
   [Identifier]

   iomega ZIP 100=Iomega 100MB Backup Device

   iomega jaz 1GB=Iomega 1GB Backup Device

   IOMEGA  ZIP 100         D.13=Iomega Corporation

   IOMEGA  ZIP 1GB         D.13=Iomega Corporation

   ...
   ```

   The format of each entry in the section is as follows:

   ```
   [Identifier]

   device_id=vendor_display_name_you_specify
   ```

   where *device_id* is the unique ID generated and updated in the registry by the vendor during the installation of the device on the inventoried server.

   For example, the contents of the section are as follows:

   ```
   [Identifier]

   iomega ZIP 100=Iomega 100MB Backup Device
   ```

   This entry is for a 100 MB Zip* drive installed on the inventoried server.

3. Add or modify the entries.

   If you specify incorrect values for the device ID entry, the device will not be displayed in the Inventory windows.

4. Click OK.

Customizing the Hardware Information for Monitor's Size

The inventory information scanned for a monitor includes the following:

Nominal Size: A number representing the diagonal width of the monitor (the distance from one corner of the screen to the opposite corner of the screen). For example, 17".

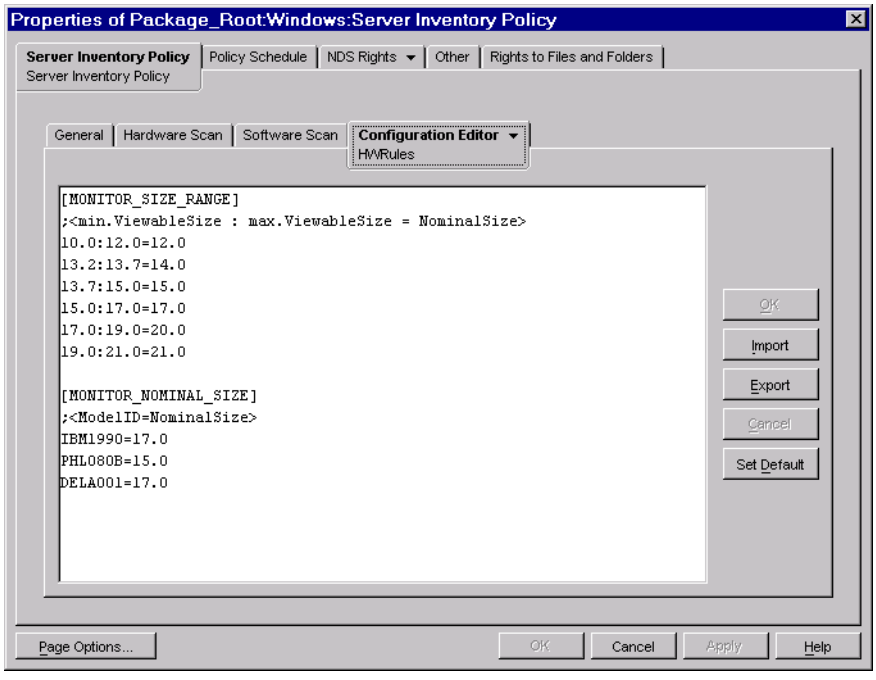
Viewable Size: A number representing the diagonal width of the screen image excluding the black borders around the image's edge. For example, 15.8".

The Inventory scanner automatically scans for the viewable size of the monitor of Windows* inventoried servers. Using the Server Inventory policy, you can customize the nominal size of the monitor to be reported.

IMPORTANT: The Inventory scanner reports inventory information for the monitors that are manufactured only after 1997.

To customize the scan of the nominal size of the monitor:

1. In the Server Inventory policy, click the Configuration Editor tab, then click HWRules.



2. Click Set Default.

   The default values are displayed in the Configuration Editor box.

3. Add or modify the entries.

   The format of HWRules.ini is as follows:

   ```
   [MONITOR_SIZE_RANGE]
   ```

   ```
   minimum_viewable_size_you_specify: maximum_viewable_size_you_specify =
   nominal_size_you_specify
   ```

   ```
   [MONITOR_NOMINAL_SIZE]
   ```

   ```
   model_ID_reported_by_scanner = nominal_size_you_specify
   ```

In the [MONITOR_SIZE_RANGE] section, specify the minimum and maximum range of the viewable size, and the corresponding nominal size of the monitor. The Inventory scanner scans for the model ID of the monitor and reports the nominal size configured in the [MONITOR_NOMINAL_SIZE] section of the HWRules.ini file.

In the [MONITOR_NOMINAL_SIZE] section, specify the model ID and its corresponding nominal size as reported by the Inventory scanner. This information is available in the Inventory Summary dialog box under the Hardware/Software Inventory > Hardware > Monitor attribute.

If the scanned model ID is not listed in [MONITOR_NOMINAL_SIZE], then the scanner scans the viewable size of the monitors. On the basis the viewable size, the scanner reports the nominal size configured in the [MONITOR_SIZE_RANGE] section of the HWRules .ini file.

For example, the contents of the HWRules .ini file could be as follows:

```
[MONITOR_SIZE_RANGE]

10.0:12.0=12.0

13.2:13.7=14.0

[MONITOR_NOMINAL_SIZE]

IBM1990=17.0

PHL080B=15.0
```

4 Click OK to save the contents to the Server Inventory policy.

Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers

The Inventory scanner has been enhanced with the following features that enables you to control the scanning process more effectively and efficiently for inventoried servers having ZENworks 6.5, ZENworks 6.5 SP1or later Server Inventory:

- It supports scanning for the following software inventory information:
  - Windows* operating system and its patches
  - Internet Explorer and its patches
  - Windows Media Player and its patches
  - Outlook Express and its patches
  - Novell Client32™ and its installed components
  - ZENworks suite and its installed components
  - Microsoft* Office and its installed applications
  - Antivirus products such as Symantec* Antivirus Corporate Edition and McAfee* Antivirus
  - Virus definition date and version for the antivirus products such as Symantec Antivirus Corporate Edition and McAfee Antivirus
- It supports scanning for the products listed in the Windows Add/Remove Programs and the MSI database.

- Includes dictionary of software titles to provide more accurate report of Installed software.
- Provides rules to control the scope of software scan.
- Reports total disk usage against configured file extensions.

This section provides information on the following topics:

What is ZENworks Software Dictionary?

ZENworks software dictionary contains a list of software identifiers and rules. Each software identifier identifies a particular product installed on an inventoried server. The rules control the scope of the scanning process.

The ZENworks software dictionary is automatically installed on an Inventory Server and inventoried servers when you install the Server Inventory software. After you configure the required policies and start the Inventory service, the Inventory scanner will report the software information on the basis of the software dictionary.

There are two types of ZENworks software dictionary: General dictionary and Private dictionary.

General Dictionary: The General dictionary is the part of the software dictionary that contains pre-defined software identifiers. On the basis of this dictionary, the Inventory scanner reports whether a particular product is installed on an inventoried server.

Private Dictionary: The private dictionary is the part of the software dictionary that contains user-defined software identifiers and rules that enable you to define the scope of Inventory scan and customize the software information. You can configure the rules. For more information on how to configure the rules, see "Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules" on page 580.

IMPORTANT: The rules that you define in the private dictionary overrides the pre-defined rules in the general dictionary.

What is a Software Dictionary Rule?

A software dictionary rule represents a set of conditions that control the scope of scanning process.

What are Software Identifiers?

An entry that identifies a software product is called as Software identifier. Each software identifier has a set of file matching attributes and corresponding software information attributes. During the Inventory scan, the scanner reads the attributes from the file headers, and if these attributes match the attributes configured in the dictionary, the information in the corresponding software information attributes is stored in the Inventory database.

What is a Key Identifier?

A software product might be identified through more than one software identifier in the dictionary. In such a scenario, the inventory scanner arbitrarily selects the software information from one of these software identifiers. A Key Identifier identifies the software identifier from which the inventory scanner should select the software information. The key identifier is useful when the different software identifiers have marginal differences between the values of the attributes (such as Description) and you want the inventory scanner to select the information from a specific software identifier.

What is an Unidentified Software?

An Unidentified software has the following characteristics:

- It is installed on the inventoried servers.
- It is configured in the Report Files with These File Extensions As Unidentified Software rule in ZENworks software dictionary.
- It is not configured in the Software Dictionary table.

What is an Inherited Rule?

An inherited rule is an entry in the software dictionary which is obtained from another Inventory server through the dictionary distribution. You cannot edit or delete these rules. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

What is An Overriding Rule?

The default software identifier in the General dictionary cannot be modified. But if you want to modify a default software identifier, you must create a new software identifier that overrides the default identifier. The inventory scanner will ignore the default the default identifier in favor of the overridden entry.

To create a software identifier that overrides a default identifier, you must specify same values for all the matching attributes defined in the default identifier and provide new values for the software information attributes.

Understanding the Usage and Precedence of ZENworks 6.5 Software Dictionary Rules

The software dictionary rules follow a precedence order. Few guidelines are applicable to all the software dictionary rules and few guidelines are applicable to certain categories of software dictionary rules. For more information, review the following sections:

- “Guidelines Applicable to All Software Dictionary Rules” on page 559
- “Precedence between Report Only Maximum Software Version and Report All Software Versions” on page 559
- “Precedence of Software Dictionary Rules Grouped in the Software Scanning Category” on page 559
- “Precedence of Software Dictionary Rules Grouped in the Disk Usage Scanning Category” on page 562

Guidelines Applicable to All Software Dictionary Rules

The following guidelines are applicable to all the software dictionary rules that you configure:

- All software dictionary rules are recorded in the software dictionary.
- All software dictionary rules are applied at the scanner.
- You can change the settings of the software dictionary rules using the Software dictionary ConsoleOne snap-ins. For more information on how to configure the software dictionary rules, see “Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 580
- Every inventory scan contains the version of dictionary files used for that scan. This information is stored in the inventory database.
- The user-defined entry overrides the default entry present in the software dictionary but only one new entry can overrides an existing entry at a time.

Precedence between Report Software with Maximum Version and Report Multiple Versions of the Software

If an entry in the “Report Multiple Versions of the Software” rule conflicts with an entry in the “Report Software with Maximum Version”, then the entry in “Report software with Maximum version” overrides the entry in the “Report Multiple Versions of the Software” rule.

Precedence of Software Dictionary Rules Grouped in the Software Scanning Category

The software dictionary rules in the Software Scanning category control the scope of scanning for the files on the local file systems.

The Software Scanning category included the following software dictionary rules:

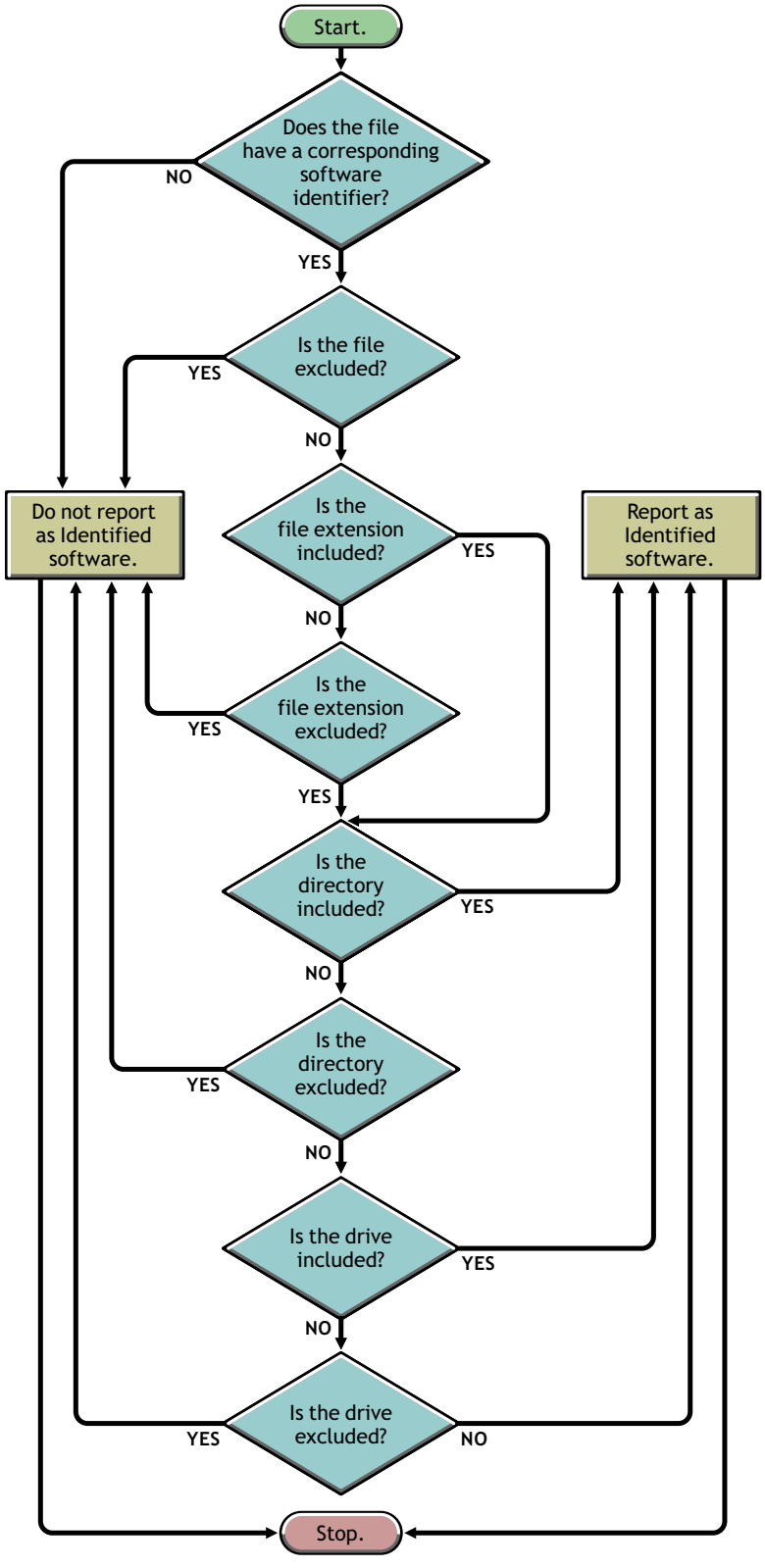
- “Include Local File Extensions” on page 571
- “Exclude Local File Extensions” on page 572
- “Include Local Directories” on page 572
- “Exclude Local Directories” on page 573
- “Include Local Drives” on page 573
- “Exclude Local Drives” on page 574
- “Exclude Local Software” on page 574
- “Exclude Local Files” on page 575

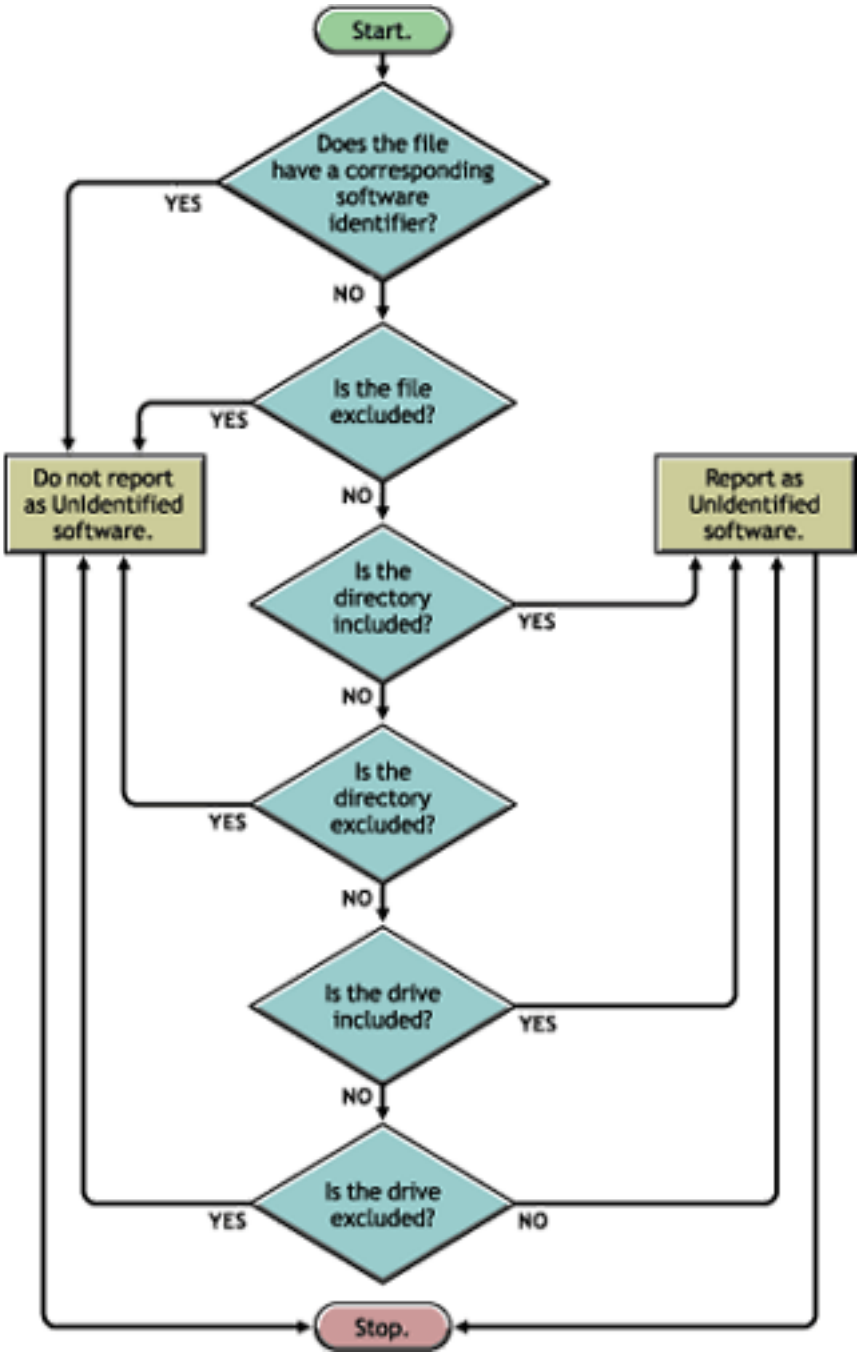
If you do not configure any of the above mentioned rules, the Inventory scanner scans for all files on the hard disk of the inventoried servers. If the files have matching entries in the software dictionary, the files are reported as identified software. If the files do not have matching entries, then they are reported as unidentified software.

If you configure the above mentioned rules, following is the precedence of rules in the descending order:

- Exclude Local Files
- Exclude Local Software
- Include Local File Extensions
- Exclude Local File Extensions
- Include Local Directories
- Exclude Local Directories
- Include Local Drives
- Exclude Local Drivers

The following flowcharts illustrate the precedence of these rules.




Start.
Does the file have a corresponding software identifier?
YES
NO
Is the file excluded?
YES
NO
Do not report as Unidentified software.
Report as Unidentified software.
Is the directory included?
YES
NO
Is the directory excluded?
YES
NO
Is the drive included?
YES
NO
Is the drive excluded?
YES
NO
Stop.

Precedence of Software Dictionary Rules Grouped in the Disk Usage Scanning Category

The software dictionary rules in the Disk Usage Scanning category determine whether a file should be considered for disk usage scan.

The Disk Usage Scanning category included the following software dictionary rules:

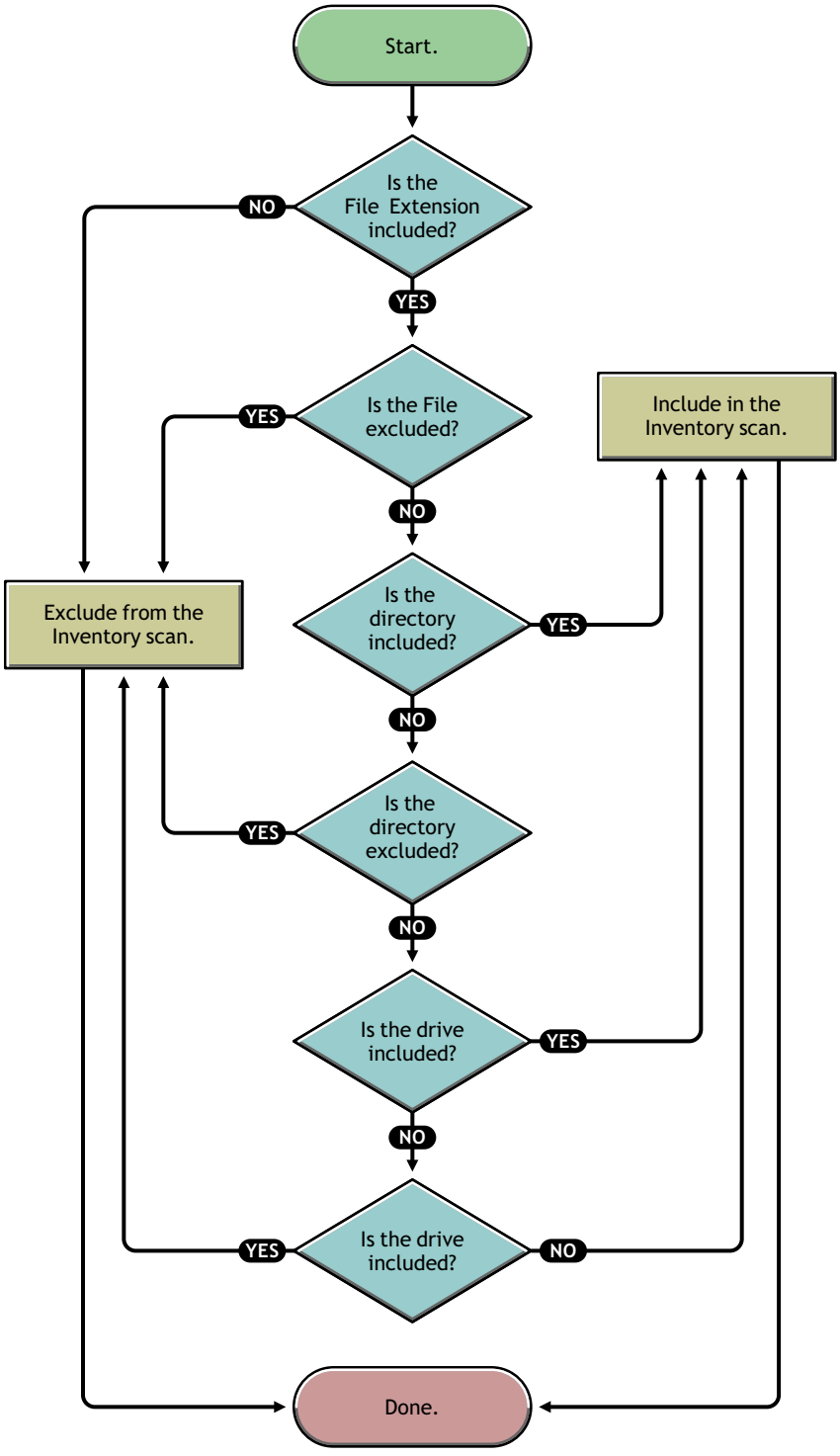
- “Disk Usage Scanning - Include Local File Extensions” on page 576
- “Disk Usage Scanning - Exclude Local Files” on page 576
- “Disk Usage Scanning - Include Local Directories” on page 576
- “Disk Usage Scanning - Exclude Local Directories” on page 577
- “Disk Usage Scanning - Include Local Drives” on page 578
- “Disk Usage Scanning - Exclude Local Drives” on page 578

For a file to be considered for the disk usage scan, its file extension must be listed in the Disk Usage Scanning - Include Local File Extensions rule and it should be excluded from inventory scan in the other Disk Usage Scanning rules.

Following is the precedence of rules in the descending order:

- Disk Usage Scanning - Exclude Local Files
- Disk Usage Scanning - Include Local Directories
- Disk Usage Scanning - Exclude Local Directories
- Disk Usage Scanning - Include Local Drives
- Disk Usage Scanning - Exclude Local Drives

The following flowchart illustrates the precedence of these rules.



Understanding the Usage and Precedence of ZENworks 6.5 SP1 or Later Software Dictionary Rules

The ZENworks software dictionary rules follow a precedence order. Few guidelines are applicable to all the software dictionary rules and few guidelines are applicable to certain categories of software dictionary rules. For more information, review the following sections:

- “Guidelines Applicable to All Software Dictionary Rules” on page 559
- “Precedence between Report Only Maximum Software Version and Report All Software Versions” on page 559
- “Precedence of Software Dictionary Rules Grouped in the Software Scanning Category” on page 559
- “Precedence of Software Dictionary Rules Grouped in the Disk Usage Scanning Category” on page 562

Guidelines Applicable to All Software Dictionary Rules

The following guidelines are applicable to all the software dictionary rules that you configure:

- All software dictionary rules are applied at the inventoried servers by the inventory scanner.
- You can change the settings of the software dictionary rules using the Software dictionary ConsoleOne snap-ins. For more information on how to configure the software dictionary rules, see “Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 580.
- Every inventory scan contains the version of dictionary files used for that scan. This information is stored in the inventory database.
- The user-defined software identifier overrides the default software identifier present in the software dictionary but at a time only one user-defined software identifier can override a default software identifier.

Precedence between Report Only Maximum Software Version and Report All Software Versions

By default, the scanner reports only the highest version of the software installed. If a rule in “Report All Software Versions” conflicts with a rule in “Report Only Maximum Software Version”, then the rule in “Report Only Maximum Software Version” overrides the rule of Report All Software Versions.

Precedence of Software Dictionary Rules Grouped in the Software Scanning Category

The software dictionary rules in the Software Scanning category control the scope of scanning for the files on the local file systems.

The Software Scanning category includes the following software dictionary rules:

- “Scan File Extensions” on page 597
- “Ignore File Extensions” on page 597
- “Scan Directories” on page 595
- “Ignore Directories” on page 594
- “Scan Drives” on page 594
- “Ignore Drives” on page 593

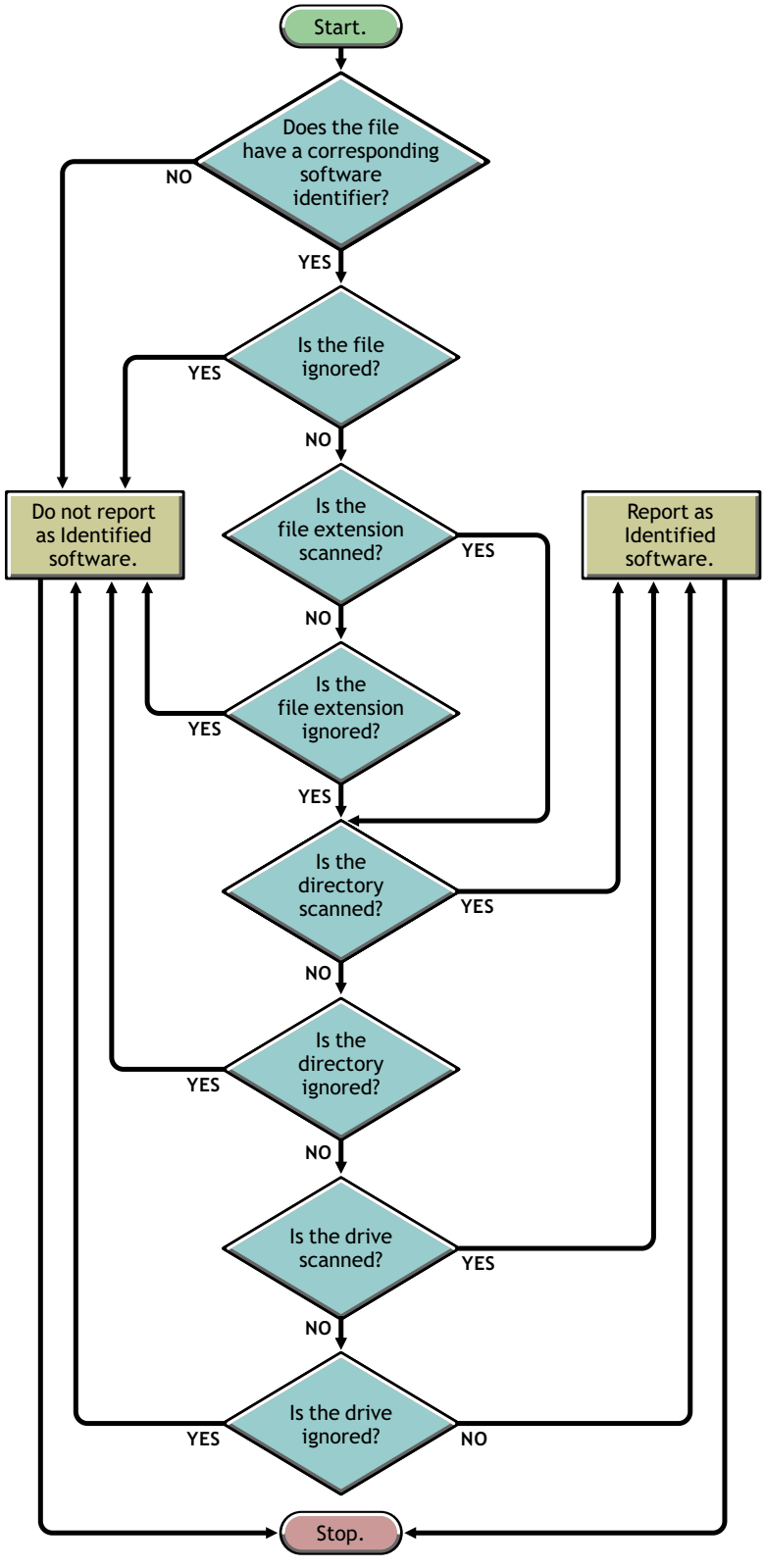
- “Software Scanning Filters - Software” on page 599
- “Software Scanning Filters - Files” on page 598

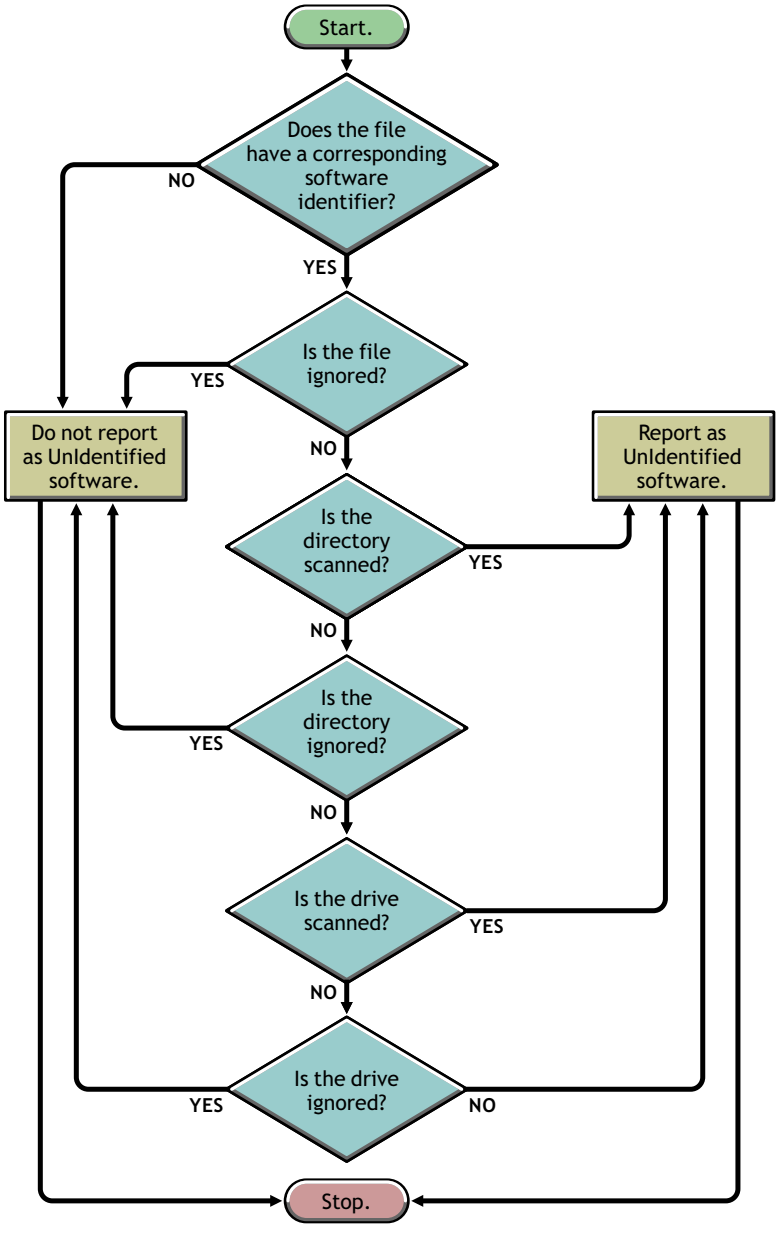
If you do not configure any of the above mentioned rules, the Inventory scanner scans for all files on the hard disk of the inventoried servers. If the files have matching software identifier in the software dictionary, the files are reported as identified software.Otherwise, they are reported as unidentified software.

If you configure the above mentioned rules, following is the precedence of rules in the descending order:

- Software Scanning Filters - Files
- Software Scanning Filters - Software
- Scan File Extensions
- Ignore File Extensions
- Scan Directories
- Ignore Directories
- Scan Drives
- Ignore Drives

The following flowcharts illustrate the precedence of these rules.





Precedence of Software Dictionary Rules Grouped in the Disk Usage Scanning Category

The software dictionary rules in the Disk Usage Scanning category determine whether a file should be considered for disk usage scan.

The Disk Usage Scanning category includes the following software dictionary rules:

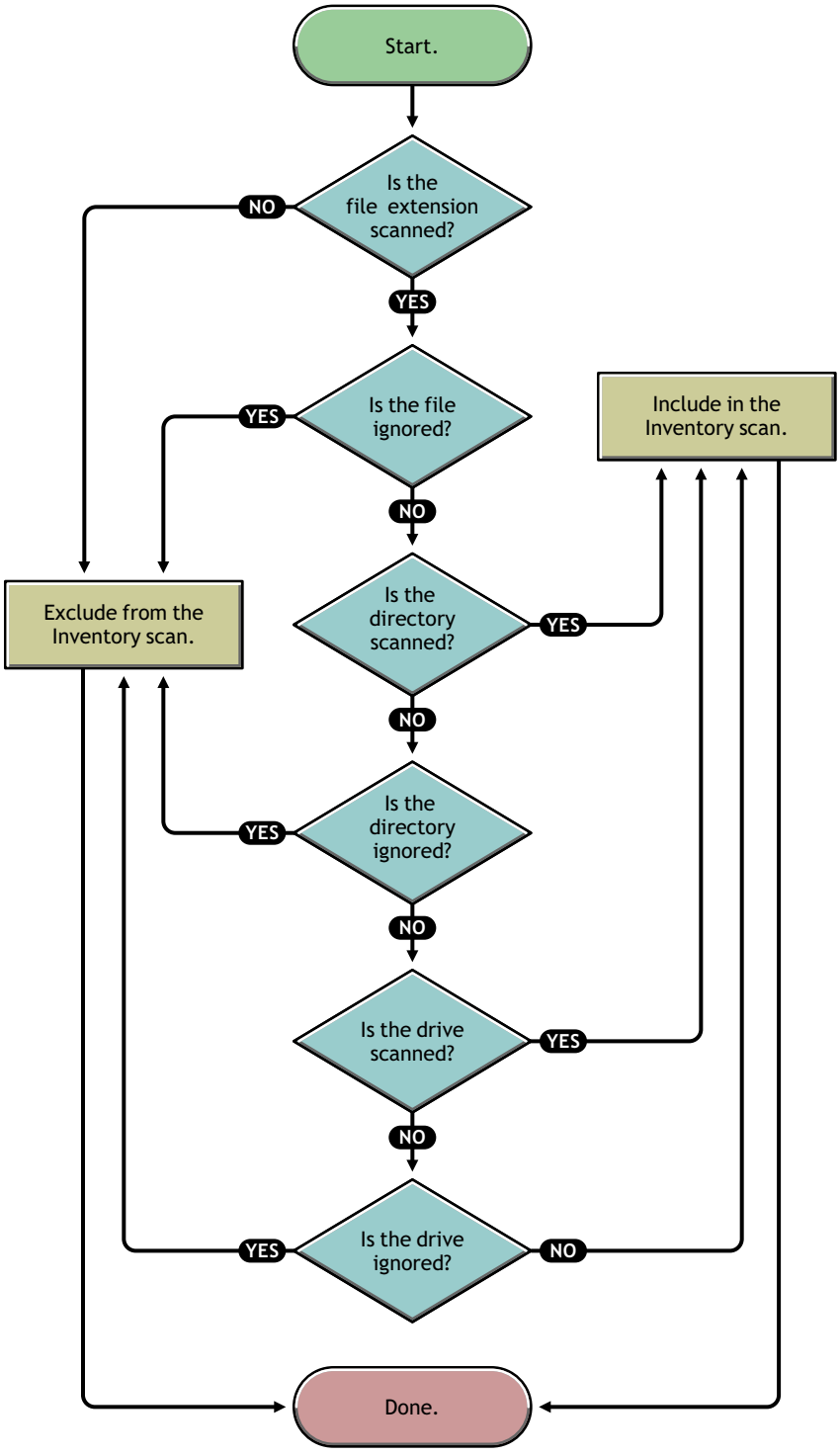
- “Report Disk Space Used by File Extensions” on page 591
- “Disk Usage Scanning Filters - Files” on page 604
- “Scan Directories” on page 604
- “Ignore Directories” on page 603
- “Scan Drives” on page 602
- “Ignore Drives” on page 602

For a file to be considered for the disk usage scan, its file extension must be listed in the “Report Disk Space Used by File Extensions” rule and it should not be excluded from inventory scan in the other Disk Usage Scanning rules.

Following is the precedence of rules in the descending order:

- Disk Usage Scanning Filters - Files
- Scan Directories
- Ignore Directories
- Scan Drives
- Ignore Drives

The following flowchart illustrates the precedence of these rules.



Understanding the Software Dictionary Pattern Types

Before configuring the software dictionary rules, you must be aware of the following software dictionary pattern types that are supported in ZENworks 6.5:

- “Regular Expression” on page 565
- “Expandable Expression” on page 565
- “System Expandable Expression” on page 565

Regular Expression

Regular Expression refers to the POSIX regular expressions. For more information on regexp (regular expressions), see The Open Group Base Specifications Issue 6 Web site (http://www.opengroup.org/onlinepubs/007904975/basedefs/xbd_chap09.html).

Examples of Regular Expression usage:

- To find all vendor names starting with “Novell,” specify Novell.*
- To find executables, specify [exe|EXE]
- To find files with name containing 6 characters, starting with “r” and ending with “t,” specify [r....t]
- To find files with name starting from A to C, and ending with E, specify [A-C].*[E]
- To find files whose name does not contain any uppercase letters, specify [^A-Z]+

NOTE: To use metacharacters such as [, \, ^, $, ., |, ?, (,), *, and + as characters, you must prefix them with a backslash (\). For example, to specify c:\windows as a regular expression, specify it as c:\\windows.

Expandable Expression

Expandable Expression contains displayable characters and the asterisk (*) wildcard character.

“*” matches to zero or more displayable characters.

Examples of Expandable Expression usage:

- To find all instances of the vendor name beginning with “Microsoft,” specify Microsoft*
- To find files with extension “.exe” in the scan, specify exe

System Expandable Expression

- **On NetWare:** A System expandable expression contains displayable characters, references to environmental variables, or the asterisk (*) wildcard character.

  IMPORTANT: “*” matches to zero or more displayable characters except when used as a unique character and is not separated by the “\” delimiter in the environmental variables. For example, in the “user*” environmental variable, “*” is a unique character; the scanner searches for the environmental variable with the name “user*”. In the “user*” environmental variable, “*” matches to zero or more displayable characters.

  Example of an environmental variable: $sysdir

- **On Windows:** A System expandable expression contains displayable characters, references to environmental variables, or the asterisk (*) wildcard character.

  “*” matches to zero or more displayable characters.

  Example of an environmental variable: %temp%

IMPORTANT: A System expandable expression can contain a combination displayable characters, references to environmental variables, or the asterisk (*) wildcard character but if it contains an environmental variable, you must specify it at the beginning of the expression. For example, %temp%/*

Examples of System Expandable Expression usage:

- To find the disk usage of the C drive, specify C
- To find files in the c:\program files directory, specify c:\program files
- To find files with extensions, ".com", specify com

Configuring the ZENworks 6.5 Software Dictionary Rules

1. In ConsoleOne, right-click the Inventory Service object (Inventory Service_*server_name*), then click Properties.
2. In the Dictionary Settings page, the Dictionary Path displays the complete path of the software dictionary files on the Inventory server. You can also use this page to access the dictionary for viewing and updating user defined entries. For more information, see "Software Dictionary" on page 583.
3. In the Software page, you can configure the following rules:
   - "Report Software with Maximum Version" on page 569
   - "Report Multiple Versions of the Software" on page 570

     IMPORTANT: Before configuring the above mentioned rules, you must be aware of the usage of these files. For more information, see "Precedence between Report Software with Maximum Version and Report Multiple Versions of the Software" on page 553.
   - "Add Unidentified Software to Dictionary" on page 570
   - "Scan as Unidentified Software" on page 571
4. In the Software Scanning page, you can configure the following rules to control the scope of scanning the files in the specified location:
   - "Include Local File Extensions" on page 571
   - "Exclude Local File Extensions" on page 572
   - "Include Local Directories" on page 572
   - "Exclude Local Directories" on page 573
   - "Include Local Drives" on page 573
   - "Exclude Local Drives" on page 574
   - "Exclude Local Software" on page 574
   - "Exclude Local Files" on page 575

   IMPORTANT: Before configuring the above listed software dictionary rules, you must be aware of the usage of these rules. For more information, see "Precedence of Software Dictionary Rules Grouped in the Software Scanning Category" on page 554.
5. In the Disk Usage Scanning page, you can configure the following disk usage information to be included and excluded during the Inventory scan:
   - "Disk Usage Scanning - Include Local File Extensions" on page 576
   - "Disk Usage Scanning - Exclude Local Files" on page 576
   - "Disk Usage Scanning - Include Local Directories" on page 576

- "Disk Usage Scanning - Exclude Local Directories" on page 577
- "Disk Usage Scanning - Include Local Drives" on page 578
- "Disk Usage Scanning - Exclude Local Drives" on page 578

IMPORTANT: Before configuring the above listed software dictionary rules, you must be aware of the usage of these rules. For more information, see "Precedence of Software Dictionary Rules Grouped in the Disk Usage Scanning Category" on page 557.

6 In the Software Aliases page, you can configure aliases for vendor names and software names. You can use this page to edit the Add-Remove software table that improves the ability of the Inventory scanner in associating the software identified through the Add Remove Programs dialog box with the corresponding software identified through the dictionary. This association is required because of the following reasons:

- The software products from the same vendor may not contain the same vendor name in the file header of the software. A software product can be identifies by different names in the dictionary and in other software information sources such as the Add-Remove Programs.
- By default, the software information is displayed for each unique vendor name in the Inventory Query window, Inventory Summary window, and the Inventory reports. If the vendor or software names differ in software dictionary and the Add-Remove Programs, the Inventory ConsoleOne utilities, display multiple entries for each vendor and software.

For more information on configuring the settings of the Software Aliases page, review the following sections:

- "Edit Aliases for Vendor Names" on page 579
- "Edit Aliases for Software Names" on page 579
- "Edit Add-Remove Software" on page 580

7 Click Apply, then click Close.

Access Software Dictionary

The "Access Software Dictionary" rule allows you to view the software dictionary entries. You can also use this table to add, modify, and delete the entries in the software dictionary. A software product can be identified by more than one software identifier. The Key Identifier option in the Access Software Dictionary table allows you to determine the key identifier whose information must be reported by the scanner.

To configure the table:

1 Click Edit Table.

2 In the Configure Software Dictionary table, click Insert to add a new row.

3 Specify values for the following identifiers.

Filename, File Date (yyyy-dd-mm), File Timestamp (hours:minutes:seconds), Minimum File Size (bytes), Maximum File Size (bytes), Product Name, Product Support Pack Version, Product Version, Internal Version, Description, Vendor, Platform, and Category.

The following identifiers are called as matching attributes: Filename, File Date, File Time, File Min Size, File Max Size, and Internal Version. From the list of matching attributes, you must configure the "File Name" attribute; others are optional. The values of these matching attributes are compared with the values scanned by the Inventory scanner from the file headers on the inventoried servers.

If the values match, then these attributes and the corresponding software information attributes (Product Name, Product Support Pack Version, Product Version, Description, Vendor, Platform, and Category) are stored in the Inventory database.

NOTE: When you add an entry in the table, a unique ID called as Dictionary Identifier is automatically assigned to this entry. You cannot change this ID.

For example, configure the following settings in the Configure Software Dictionary table:

Filename= MSACCESS.EXE
File Date= 1998-30-01
File Timestamp= 05:30
Minimum File Size= 299854
Maximum File Size= 400000
Product Name= Access
Product Version= 7.0
Internal Version= 7.0
Description= Microsoft Access
Vendor= Microsoft
Platform= win95
Category= Database

If the Inventory scanner finds a file with the following values during the scan: “File Name= MSACCESS.EXE; File Date= 1998-30-01; File Timestamp= 05:30; File Size= 400000; Internal Version= 7.0,” then the following information is stored in the Inventory database:

File Name= MSACCESS.EXE
File Date= 1998-30-01
File Timestamp= 05:30
File Size= 400000
Product Name= Access
Product Version= 7.0
Internal Version= 7.0
Description= Microsoft Access
Vendor= Microsoft
Platform= win95
Category= Database

IMPORTANT: When you add a new entry to the table, ensure that the value of one of its matching attribute is unique.

4 (Optional) Select the Key Identifier check box for this entry.

A key identifier helps you to identify a software product. Each key identifier has a set of file matching attributes and corresponding software information attributes. During the Inventory scan, the scanner reads the attributes from the file headers, and if these attributes match the attributes configured in the dictionary, the information in the corresponding software information attributes is stored in the Inventory database.

For example, the Configure Dictionary table has the following entries for MS Word:

Filename	File date	File timestamp	Minimum File Size	Maximum File Size	Product name	Product version	Internal version	Description	Vendor
winword.exe	2004-30-1	05:30	10000	10000	Word	2002	10.0.4219	Microsoft Word	Microsoft
osa.exe	2004-30-02	16:00	10000	10000	Word	2002	10.0.4300	Microsoft Office XP Component	Microsoft

If the key identifier has not been defined, the software information for MS Word might be selected from anyone of the above entries.

To ensure that the information from the identifier corresponding to “Winword.exe” is selected, you must select the key identifier check box for “Winword.exe.” If you select “Winword.exe” as the key identifier in the Configure Software Dictionary table, the Inventory scanner will store the information related to winword.exe into the Inventory database.

5 Click OK.

IMPORTANT: You can automatically add entries to this table from the Configure Scanned Unidentified Software table located in the Add Unidentified Software to Dictionary rule.

In the Configure Software Dictionary table, you can also perform the following tasks:

- Exclude a software from scan by adding the software to the table in the Exclude Local Software rule.

  In the Configure Software Dictionary table, select the software entry that you want to exclude during the next scan, and click the Exclude Software button.
- Sort the entries of the table.
- Delete the entries from the table.
- Apply filters to the entries of the table.

Report Software with Maximum Version

The “Report Software with Maximum Version” rule allows you to configure the software, whose highest version must be reported by the Inventory scanner.

To configure the rule:

1 Click Edit Table.

2 In the table, click Insert to add a new row.

3 In the Name Pattern drop-down list, select Expandable Expression or Regular Expression.

4 Specify a software name pattern.

5 (Optional) In the Vendor Pattern drop-down list, select Expandable Expression or Regular Expression.

6 (Optional) Specify a vendor name pattern.

7 Click OK.

For example, if you want the Inventory scanner to report only the highest version of the Adobe Acrobat Reader installed on the inventoried server, configure the following settings in the table:

Name Pattern = Expandable expression
Name= Acrobat* Reader*
Vendor Pattern = Expandable expression
Vendor = Adobe*

If the inventoried server has Adobe Acrobat Reader versions 4.0 and 5.0 installed, the Inventory scanner reports Adobe Acrobat Reader 5.0.

NOTE: By default, the scanner reports only the highest version of the software.

Report Multiple Versions of the Software

The "Report Multiple Versions of the Software" rule allows you to configure the software for which the Inventory scanner must report all versions of the software installed on the inventoried server.

To configure the rule:

1. Click Edit Table.
2. In the table, click Insert to add a new row.
3. In the Name Pattern drop-down list, select Expandable Expression or Regular Expression.
4. Specify a software name pattern.
5. In the Vendor Pattern drop-down list, select Expandable Expression or Regular Expression.
6. Specify a vendor name.
7. Click OK.

For example, if you want the Inventory scanner to report all versions of the Adobe Acrobat Reader installed on the inventoried server, configure the following settings in the table:

Name Pattern = Expandable expression
Name= Acrobat* Reader*
Vendor Pattern = Expandable expression
Vendor = Adobe*

If the inventoried server has Acrobat Reader 6.0 and Acrobat Reader 5.0 installed, the Inventory scanner will report Acrobat Reader 6.0 and Acrobat Reader 5.0.

NOTE: If you configure a rule that conflicts with a rule in Report Software with Maximum Version, then the rule in the "Report software with maximum version" will override the new rule.

Add Unidentified Software to Dictionary

The software dictionary might not identify all the software products installed in you network. The software that are not listed in the dictionary are called Unidentified software.

Click Edit table. The Configure Scanned Unidentified Software table lists the unidentified software that is stored in the Inventory database.

If you want the unidentified software to be identified in subsequent scans, select the software entry in the table and click the Software Dictionary button. The entry is automatically added to Configure Software Dictionary table located in the Software Dictionary rule.

In the Configure Scanned Unidentified Software table, you can also sort the entries, and also apply filters to the entries of the table.

Scan as Unidentified Software

Allows you to configure the file extensions of the files, that should included in the list of "Unidentified" software. You can view the list of unidentified software in the Add Unidentified Software to Dictionary rule.

To configure the rule:

1. Click Edit Table.
2. In the Configure Software to be Reported as Unidentified table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a file extension pattern.
5. Click OK.

For example, if you want the Inventory scanner to report all files with extension ".exe" that do not have an entry in the dictionary as "Unidentified," configure the following settings in the table:

Pattern Type = Expandable expression
File Extension = exe

Include Local File Extensions

The "Include Local File Extensions" rule allows you to configure the files with specified extensions that should be scanned for at the inventoried servers. This rule acts as a filter on the list of software identifiers. A file that is listed as a software identifier and whose extension is included in this table will be reported

To configure a file extension to be included in the scan:

1. Click Edit Table.
2. In the File Extension Include table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a file extension pattern.
5. Click OK.

For example, if you want the Inventory scanner to scan for files with extension ".exe," configure the following settings in the Configure File Extension Includes table:

Pattern Type = Regular expression
File Extension = [exe|EXE]

The Inventory scanner scans and stores only the files with extension "exe" in the Inventory database.

NOTE: You must exclude all other file extensions present on the machine for this rule to be effective by configuring the Exclude Local File Extensions rule.

Exclude Local File Extensions

The “Exclude Local File Extensions” rule allows you to configure the file with specified extensions that should not be scanned for at the inventoried servers.

To configure a file extension to be excluded from the scan:

1. Click Edit Table.
2. In the Configure File Extension Exclude table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a file extension expression
5. Click OK.

For example, if you do not want the Inventory scanner to scan for files with extension “.exe,” configure the following settings in the Configure File Extension Excludes table:

Pattern Type = Expandable expression
File Extension = exe

The Inventory scanner will not scan for the files with the “.ext” extension.

IMPORTANT: Since the default behavior of the scanner is to include all file extensions, you need not configure the Include Local File Extensions rule for the remaining file extensions on the inventoried server.

Include Local Directories

The “Include Local Directories” rule allows you to configure the directories that should be scanned for files at the inventoried servers.

To configure a directory to be included in the scan:

1. Click Edit Table.
2. In the Configure Scan - Include Directories table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows, or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a directory name expression.
6. Click OK.

For example, if you want the Inventory scanner to scan for files in the c:\programfiles directory on all the Windows inventoried servers, configure the following settings in the Configure Scan - Include Directories table:

Platform = Windows
Pattern Type = System expandable expression
Directory= C:\ProgramFiles

The Inventory scanner scans only the files in c:\programfiles for software information.

NOTE: You must exclude all other directories present on the machine for this rule to be effective by configuring the Exclude Local Directories rule.

Exclude Local Directories

The “Exclude Local Directories” rule allows you to configure the directories that should not be scanned for files at the inventoried servers.

To configure a directory to be excluded during the scan:

1. Click Edit Table.
2. In the Configure Scan - Exclude Directories table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows, or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a directory name pattern.
6. Click OK.

For example, if you want the Inventory scanner not to scan the files in the c:\programfiles directory on all the Windows inventoried servers, configure the following settings in the Configure Scan - Exclude Directories table:

Platform = Windows
Pattern Type = System expandable expression
Directory= C:\ProgramFiles

The Inventory scanner will not scan for the files in C:\ProgramFiles.

IMPORTANT: Since the default behavior of the scanner is to include all directories, you need not configure the Include Local Directories rule for the remaining directories on the inventoried server.

Include Local Drives

The “Include Local Drives” rule allows you to configure the drives or volumes that should be scanned for files at the inventoried servers.

To configure a drive to be included in the scan:

1. Click Edit Table.
2. In the Configure Include Drive table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows, or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a drive name pattern.
6. Click OK.

For example, if you want the Inventory scanner to scan for files in the C: drive on all the Windows inventoried servers, configure the following settings in the Configure Include Drives table:

Platform = Windows
Pattern Type = System expandable expression
Drive Name= C

The Inventory scanner scans only the files in the C drive for the software information.

NOTE: You must exclude all other drives present on the machine for this rule to be effective by configuring the Exclude Local Drives rule.

Exclude Local Drives

The "Exclude Local Drives" rule allows you to configure the drives that should not be scanned for files at the inventoried servers.

To configure a drive to be excluded during the scan:

1. Click Edit Table.
2. In the Configure Scan - Exclude Drives table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to "Any." If you select System expandable expression, you cannot select "Any" as the platform.
5. Specify a drive name pattern.
6. Click OK.

For example, if you want the Inventory scanner not to scan the files in C: drive on all the Windows inventoried servers, configure the following settings in the Configure Scan - Exclude Drives table:

Platform = Windows
Pattern Type = System expandable expression
Drive Name= C

The Inventory scanner will not scan the files in the C drive.

IMPORTANT: Since the default behavior of the scanner is to include all drives, you need not configure the Include Local Drives rule for the remaining drives on the inventoried server.

Exclude Local Software

The "Exclude Local Software" rule allows you to configure the software that should not be reported by the Inventory scanner.

To configure a software not to be reported:

1. Click Edit Table.
2. In the Configure Exclude Software table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a software name pattern.
5. Click OK.

For example, if you want the Inventory scanner not to report any software whose name begin with “Adobe”, configure the following settings in the Configure Exclude Software table:

Pattern Type= Expandable expression

Software = Adobe*

The Inventory scanner will not report the software with name beginning with “Adobe.”

You can also add the entries to table in the following ways:

- Click the Add from Database button in the Configure Exclude Software table. The Add from Database option allows you to choose the software that has to be excluded from the list of software reported by the scanner.

  The Exclude Scanned Software table lists the software that is stored in the Inventory database. If you want a software product to be excluded from subsequent scans, select the software entry in the table and click Exclude Software. The entry is added to the Configure Exclude Software table located in the Exclude Local Software rule.

  In the Scanned Software in Database list, you can sort the entries and also, apply filters to the entries.

- In the Configure Software Dictionary table (located in the Software Dictionary rule), select the software that you want to exclude during the next scan, and click the Exclude Software button.

Exclude Local Files

The “Exclude Local Files” rule allows you to configure the files that should be excluded during the Inventory scan. These files will be skipped by the Inventory scanner.

To configure the software:

1. Click Edit Table.
2. In the Exclude Local Files table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a filename pattern
6. Click OK.

For example, if you do not want the Inventory scanner to report files with extension, “.com” from Windows inventoried servers, configure the following settings in the Exclude Local Files table:

Platform= Windows

Pattern Type = System expandable expression

Software = *.com

Disk Usage Scanning - Include Local File Extensions

The “Include Local File Extensions” rule allows you to configure the files with specified extensions whose disk usage should be scanned for at the inventoried servers.

1. Click Edit Table.
2. In the Configure Usage - File Extensions table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a file extension pattern.
5. Click OK.

For example, if you want the Inventory scanner to scan for disk usage of all files with extension “.mp3,” configure the following settings in the Configure Usage - File Extensions table:

Pattern Type = Expandable expression
File Extension = mp3

The Inventory scanner considers only the files with extension “.mp3” for disk usage scanning.

Disk Usage Scanning - Exclude Local Files

The “Exclude Local Files” rule allows you to configure a file whose disk usage should not be scanned for at the inventoried servers.

1. Click Edit Table.
2. In the Configure Usage - Ignore Files table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a filename extension.
6. Click OK.

For example, if you want the Inventory scanner not to scan for the disk usage of all executables whose name start with “gwe,” configure the following settings in the Configure Usage - Ignore Files table:

Platform = Windows
Pattern Type = System expandable expression
File Extension = gwe*.exe

The Inventory scanner will not consider the files of “gwe*.ex” pattern for disk usage scanning.

Disk Usage Scanning - Include Local Directories

The “Include Local Directories” rule allows you to configure the directories on the inventoried servers to be included in the disk usage scanning.

1. Click Edit Table.
2. In the Configure Usage - Include Directories table, click Insert to add a new row.

3 In the Platform drop-down list, select NetWare, Windows or Any.

4 In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.

5 Specify a directory name pattern.

6 Click OK.

For example, if you want the Inventory scanner to include the c:\programfiles directory on all the Windows inventoried servers in disk usage scanning, configure the following settings in the Configure Usage - Include Directories table:

Platform = Windows
Pattern Type = System expandable expression
Directory= c:\programfiles

The Inventory scanner considers only the files in the c:\programfiles for disk usage scanning.

NOTE: You must exclude all other directories present on the inventoried server for this rule to be effective by configuring the Disk Usage Scanning - Exclude Local Directories rule.

Disk Usage Scanning - Exclude Local Directories

The “Exclude Local Directories” rule allows you to configure the directories on the inventoried servers that should be excluded from disk usage scanning.

1 Click Edit Table.

2 In the Configure Usage - Exclude Directories table, click Insert to add a new row.

3 In the Platform drop-down list, select NetWare, Windows or Any.

4 In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.

5 Specify a directory name pattern.

6 Click OK.

For example, if you want the Inventory scanner to exclude the c:\programfiles directory on all the Windows inventoried servers from disk usage scanning, configure the following settings in the Configure Usage - Exclude Directories table:

Platform = Windows
Pattern Type = System expandable expression
Directory= c:\programfiles

The Inventory scanner will not scan for the disk usage of C:\ProgramFiles.

IMPORTANT: Since the default behavior of the scanner is to include all directories, you need not configure the Disk Usage Scanning - Include Local Directories rule for the remaining directories on the inventoried servers.

Disk Usage Scanning - Include Local Drives

The “Include Local Drives” rule allows you to configure the drives or volumes on the inventoried servers to be considered for disk usage scanning.

1. Click Edit Table.
2. In the Configure Usage - Include Drive table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a drive name pattern.
6. Click OK.

For example, if you want the Inventory scanner to scan for the disk usage of C drive on all the Windows inventoried servers, configure the following settings in the Configure Usage - Include Drive table:

Platform = Windows
Pattern Type = System expandable expression
Drive Name= C

The Inventory scanner considers the files in C drive on the Windows inventoried servers for disk usage scanning.

NOTE: You must exclude all other drives present on the inventoried server for this rule to be effective by configuring the Disk Usage Scanning - Exclude Local Drives rule.

Disk Usage Scanning - Exclude Local Drives

The “Exclude Local Drives” rule allows you to configure the drives or volumes on the inventoried servers that should be excluded from disk usage scanning.

1. Click Edit Table.
2. In the Configure Usage - Exclude Drives table, click Insert to add a new row.
3. In the Platform drop-down list, select NetWare, Windows or Any.
4. In the Pattern Type drop-down list, select System Expandable Expression or Regular Expression.

   IMPORTANT: If you select Regular expression as the pattern type, then the corresponding value in the Platform column is automatically changed to “Any.” If you select System expandable expression, you cannot select “Any” as the platform.
5. Specify a drive name pattern.
6. Click OK.

For example, if you want the Inventory scanner not to scan for the disk usage of C: on all the Windows inventoried servers, configure the following settings in the Configure Usage - Exclude Drives table:

Platform = Windows

Pattern Type = System expandable expression

Drive Name= C

The Inventory scanner will not consider the files in C drive on the Windows inventoried servers for disk usage scanning.

IMPORTANT: Since the default behavior of the scanner is to include all drives and volumes, you need not configure the Disk Usage Scanning - Include Local Drives rule for the remaining drives or volumes of the inventoried server.

Edit Aliases for Vendor Names

The “Aliases for Vendor Names” rule allows you to configure aliases for vendor names.

To configure an alias for a vendor name:

1. Click Edit Table.
2. In the Edit Vendor Name Aliases table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a vendor name pattern.
5. Specify an alias.
6. Click OK.

For example, if you want the Inventory scanner to report all instances of the vendor name beginning with “Microsoft” as “Microsoft Corporation” in the Inventory database, configure the following settings in the Edit Vendor Name Aliases table:

Pattern Type = Expandable expression

Pattern = Microsoft*

Alias = Microsoft Corporation

If the Inventory scanner reports the following vendor names during the scan: Microsoft, Microsoft Inc., Microsoft Inc. Corporation; then all these names are reported as “Microsoft Corporation” in the Inventory database.

Edit Aliases for Software Names

The “Aliases for Software Names” rule allows you to configure aliases for the software names.

To configure an alias for a software name:

1. Click Edit Table.
2. In the Edit Software Name Aliases table, click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable Expression or Regular Expression.
4. Specify a software name pattern.
5. Specify an alias.
6. Click OK.

For example, if you want the Inventory scanner to report all instances of the product name, “WinZip” as “WinZip Application” in the Inventory database, configure the following settings in the Edit Software Name Aliases table:

Pattern Type = Expandable expression

Pattern = WinZip

Alias = WinZip Application

If the Inventory scanner scans the following product names during the scan: WinZip, WinZip Executables, WinZip Applications; then the name of the software that exactly match “WinZip” is stored as “WinZip Application” in the Inventory database. The names of the remaining software are reported as scanned.

Edit Add-Remove Software

The “Edit Add-Remove Software” rule improves the ability of the scanner in associating the software identified through the Add Remove Programs dialog box with the corresponding software identified through the dictionary.

Click Edit Table. By default, the Edit Add-Remove Software table displays the following information that has been stored in the database:

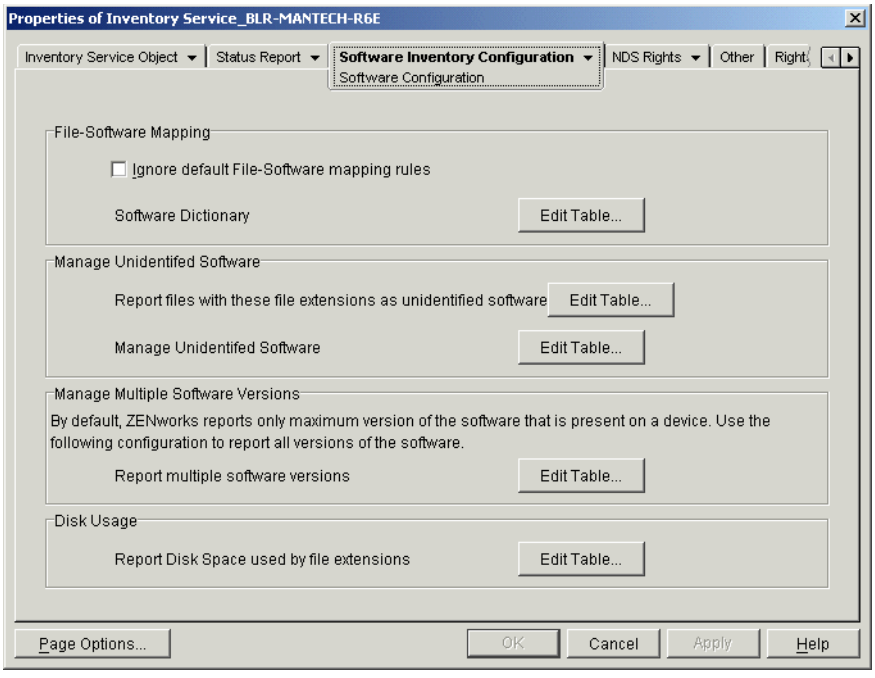
- Software key (list of registry keys for a software)
- Displayed software name (the name of the software as displayed in the Add-Remove Programs dialog box)

You can specify the software name and the vendor name that matches with the corresponding value in the dictionary.

Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules

1. In ConsoleOne, right-click the Inventory Service object (Inventory Service_*server_name*), then click Properties.
2. In the Software Configuration page, you can configure the following settings to scan the software inventory information:
   - **File - Software Mapping:** Includes the following rules:
     - “Ignore Default File-Software Mapping Rules” on page 583
     - “Software Dictionary” on page 583
   - **Manage Unidentified Software:** Includes the following rules:
     - “Report Files with These File Extensions As Unidentified Software” on page 587
     - “Manage Unidentified Software” on page 588
   - **Manage Multiple Software Versions:** Includes the following rule:
     - “Report Multiple Software Versions” on page 589
   - **Disk Usage:** Includes the following rule:
     - “Report Disk Space Used by File Extensions” on page 591

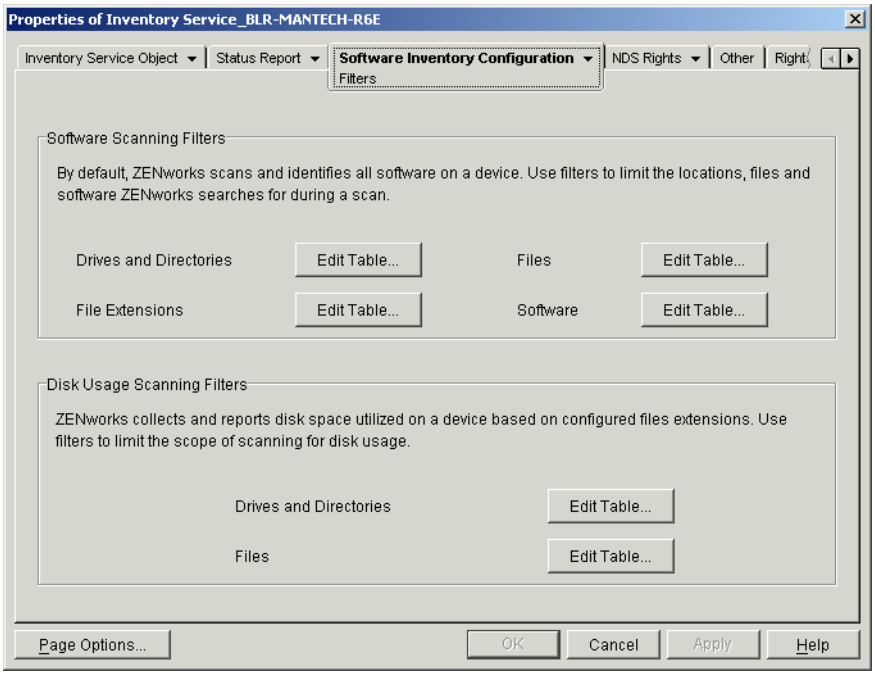
   IMPORTANT: Before configuring any ZENworks software dictionary rules, you must be aware of how to use these rules. For detailed information, see “Understanding the Usage and Precedence of ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 559.



3 In the Filters page, you can control the scope of scanning for files.

- **Software Scanning Filters:** Includes the following filters:
  - "Software Scanning Filters - Drives and Directories" on page 592
  - "Software Scanning Filters - File Extensions" on page 596
  - "Software Scanning Filters - Files" on page 598
  - "Software Scanning Filters - Software" on page 599
- **Disk Usage Scanning Filters:** Includes the following filters:
  - "Disk Usage Scanning Filters - Drives and Directories" on page 600
  - "Disk Usage Scanning Filters - Files" on page 604

IMPORTANT: Before configuring any ZENworks software dictionary rules, you must be aware of how to use these rules. For detailed information, see "Understanding the Usage and Precedence of ZENworks 6.5 SP1 or Later Software Dictionary Rules" on page 559.



4 In the Aliases page, you can do the following:

- **Specify Aliases:** Allows you to configure aliases for vendor and software names.

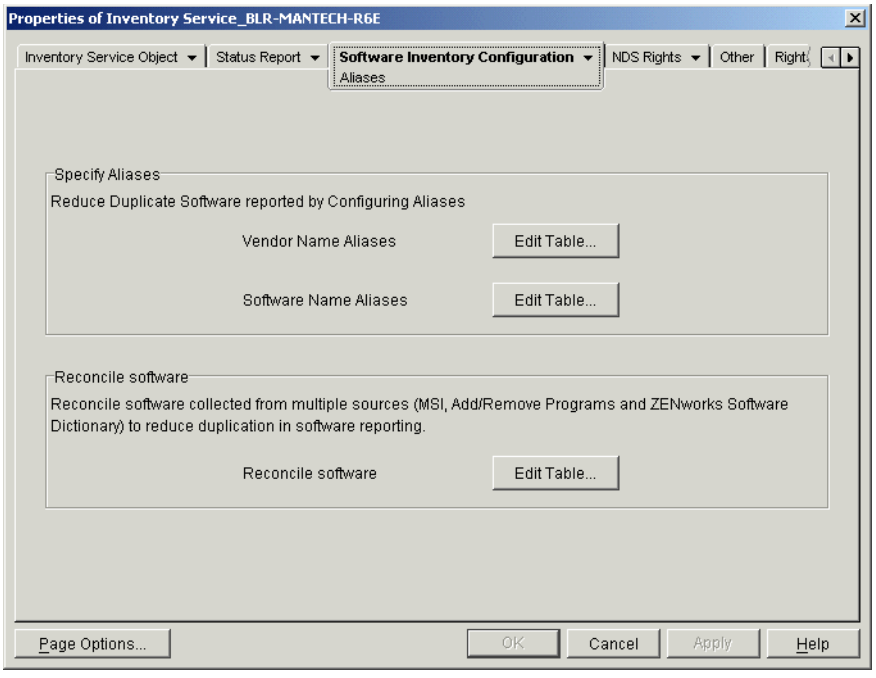
  By default, the software information is categorized by vendor name in the Inventory ConsoleOne utilities. The software from the same vendor might sometimes have differing vendor names or product names. In this scenario, the Inventory ConsoleOne utilities display the software information under different sections.

  However, you can merge the software information by specifying aliases. You customize these settings in the following software dictionary rules:

  - “Vendor Name Aliases” on page 606
  - “Software Name Aliases” on page 607

- **Reconcile Software:** Allows you to merge the software identified through Add/Remove Programs or the MSI, and the software identified through the ZENworks software dictionary. For more information, see “Reconcile Software” on page 608.

IMPORTANT: Before configuring any ZENworks software dictionary rules, you must be aware of how to use these rules. For detailed information, see “Understanding the Usage and Precedence of ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 559.



5 Click Apply, then click Close.

Ignore Default File-Software Mapping Rules

Select the “Ignore Default File-Software Mapping Rules” check box if you do not want the Inventory scanner to use the default File-Software mapping rules that are configured in the ZENworks software dictionary for scanning software inventory information.

IMPORTANT: This option is not available for selection if the software dictionary is updated from another Inventory server.

Software Dictionary

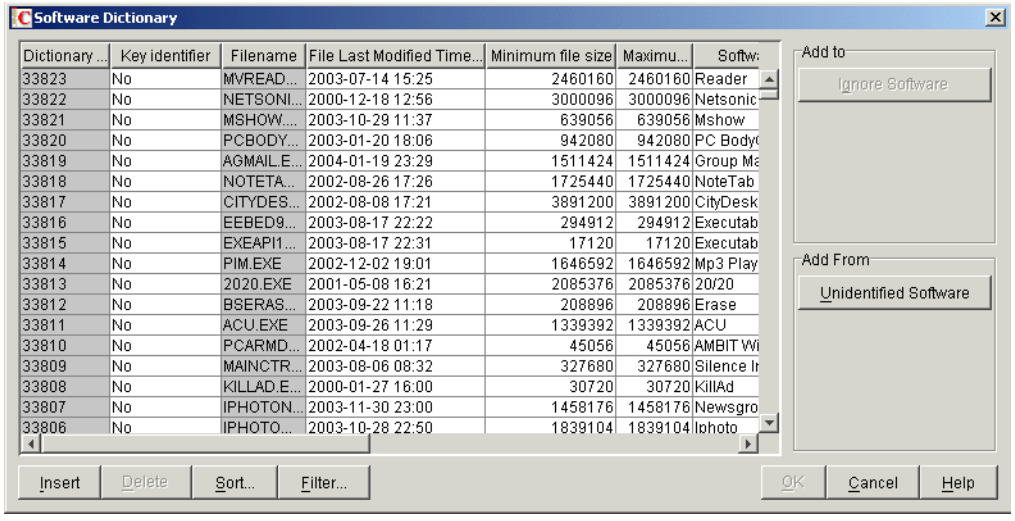
The “Software Dictionary” option allows you to configure software identifiers in the ZENworks software dictionary.

By default, the ZENworks software dictionary contains predefined software identifiers. You can create new software identifiers in the ZENworks software dictionary by editing the predefined software identifiers or creating a new software identifier.

To configure rules in the ZENworks software dictionary:

1 Click Edit Table.

The Software Dictionary table is displayed.



The Software Dictionary table displays the data stored in the ZENworks software dictionary. It might contain entries that are:

- **Light gray in color:** Indicates that these entries will not be considered in a scan because the table already contains entries that override these entries.
- **Dark gray in color:** These are inherited rules. For more information about inherited rules, see "What is an Inherited Rule?" on page 552.

2 In the Software Dictionary table, you can perform the following operations:

- "Manually Adding Entries to the Software Dictionary" on page 584
- "Automatically Adding Entries to the Software Dictionary" on page 586
- "Deleting Entries from the Software Dictionary" on page 586
- "Modifying the Values of the Software Dictionary Entries" on page 586
- "Excluding a Software from a Scan" on page 587
- "Sorting Entries in the Table" on page 609
- "Filtering Entries in the Table" on page 609
- "Refreshing Entries in the Table" on page 610

3 Click OK.

Manually Adding Entries to the Software Dictionary

1 In the Software Dictionary table, click Insert to add a new row.

2 Specify values for the following attributes:

Filename, File Last Modified Time (yyyy-dd-mm hours:minutes), Minimum File Size (bytes), Maximum File Size (bytes), Software Name, Support Pack, Software Version, Internal Version, Description, Vendor, Platform, and Category.

The following attributes are called "matching attributes": Filename, File Last Modified Time, Minimum File Size, Maximum File Size, and Internal Version. The values of these matching attributes are compared with the values scanned by the Inventory scanner from the file headers on the inventoried servers.

If the values are same, the values in the corresponding software information attributes (Software Name, Support Pack, Software Version, Description, Vendor, Platform, and Category) are stored in the Inventory database.

In the Software Dictionary table, you must specify values for the following attributes: Filename, Software Name, and Vendor. It is optional to specify values for other attributes.

When you add an entry, a unique ID called the Dictionary Identifier is automatically assigned to this entry

For example, configure the following settings in the Software Dictionary - Row Editor table:

Filename= MSACCESS.EXE
File Last Modified Time = 1998-30-01 05:30
Minimum File Size = 299854
Maximum File Size = 400000
Software Name = Access
Software Version = 7.0
Internal Version = 7.0
Description = Microsoft Access
Vendor = Microsoft
Category = Database

If the Inventory scanner finds a file with the following values during the scan: “File Name= MSACCESS.EXE; File Last Modified Time= 1998-30-01 05:30; File Size= 300000,” then the following information is stored in the Inventory database:

Software Name = Access
Software Version = 7.0
Description = Microsoft Access
Vendor = Microsoft
Category = Database

If you do not specify a value for an attribute, then this attribute is not considered to determine the overriding entry. Also, only the matching attributes are considered to determine the overriding entry. For example, the Configure Dictionary table has the following entries for MS Word:

Filename	Minimum File Size	Maximum File Size	Software Name	Vendor
winword.exe	10000	10000	Word	Microsoft
winword.exe	0	30000	Word	Microsoft

To determine the overriding entry, only the maximum file size value is considered. Consequently, the second entry with 30000 maximum file size overrides the first entry.

3 (Optional) Select the Key Identifier check box for this entry.

For example, the Software Dictionary table has the following entries for MS Word:

Filename	File Last Modified Time	Minimum File Size	Maximum File Size	Software Name	Software version	Internal version	Description	Vendor
winword.exe	2004-30-10 5:30	10000	10000	Word	2002	10.0.4219	Microsoft Word	Microsoft
osa.exe	2004-30-02 16:00	10000	10000	Word	2002	10.0.4300	Microsoft Office XP Component	Microsoft

If the key identifier has not been defined, the software information for MS Word might be selected from anyone of the above entries.

To ensure that the information from the identifier corresponding to “Winword.exe” is selected, select Key Identifier for “Winword.exe.” If you select “Winword.exe” as the key identifier in the Configure Software Dictionary table, the Inventory scanner stores the information related to winword.exe into the Inventory database.

Automatically Adding Entries to the Software Dictionary

1 Click Unidentified Software located in the Add From pane.

2 In the Manage Unidentified Software table, do the following:

2a Select the entry to be added to the software dictionary.

2b Click Software Dictionary located in the Add To pane.

2c Click Close.

Deleting Entries from the Software Dictionary

1 Select the entry to be deleted.

2 Click Delete.

IMPORTANT: You can delete only the non-inherited entries.

Modifying the Values of the Software Dictionary Entries

1 In the Software Dictionary table, double-click the entry whose values you want to modify.

You can modify only one entry at a time.

TIP: You can also invoke the Row Editor dialog box by selecting the entry you want to modify and pressing either one of the keys: Enter, Spacebar, or F2.

2 Modify the values.

You cannot modify the values of the Dictionary Identifier and Filename attributes.

3 Click OK.

IMPORTANT: You cannot modify the values of an inherited rule. Also, modifying a default predefined rule creates a new user-defined rule.

Excluding a Software from a Scan

1. In the Software Dictionary table, select the corresponding entry for the software that must be excluded from the Inventory scan.
2. Click Ignore Software located in the Add To pane.

   The entry is added to the Ignore Software table in Software Scanning Filters - Software.

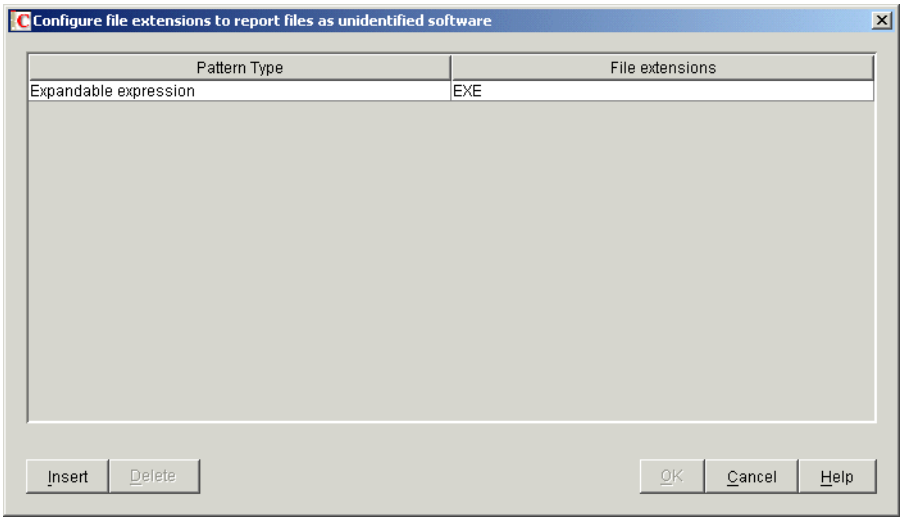
Report Files with These File Extensions As Unidentified Software

The “Report Files with These File Extensions As Unidentified Software” rule allows you to configure file extension of files that must be reported as unidentified software.

To configure the rule:

1. Click Edit Table.

   The “Configure File Extensions to Report Files as Unidentified Software” table is displayed.



2. Click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable expression or Regular expression.
4. Specify a file extension.
5. Click OK.

For example, if you want the Inventory scanner to report the software with the “.exe” extension as Unidentified software, configure the following settings in the table:

Pattern Type = Expandable expression

File Extensions = exe

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1or later Inventory server.

To delete an entry from the table, select the entry and click Delete. You can delete only the non-inherited entries.

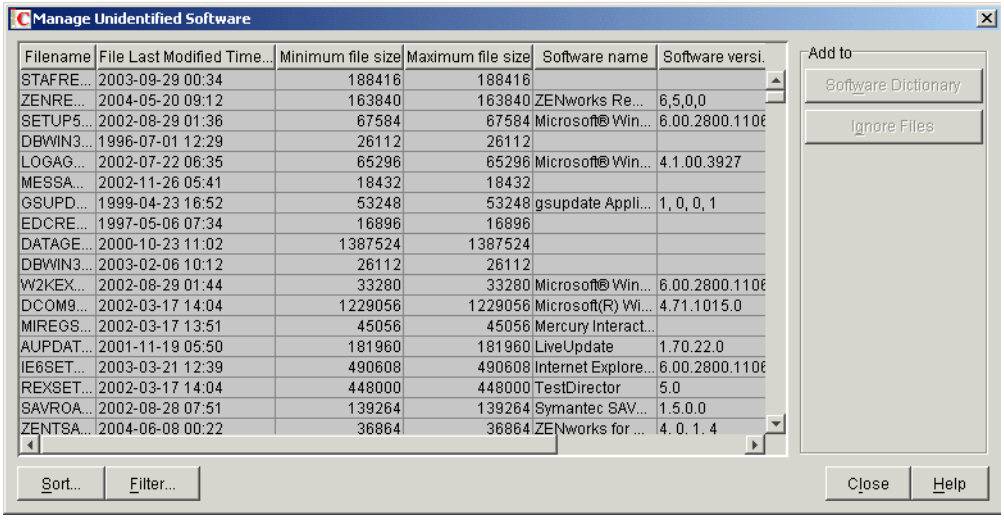
Manage Unidentified Software

The “Manage Unidentified Software” rule allows you to include or exclude the unidentified software from the inventory scan.

To configure this rule:

1 Click Edit Table.

The Manage Unidentified Software table is displayed.



IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

2 In the Manage Unidentified Software table, you can perform the following operations:

- “Including Unidentified Software in a Scan” on page 588
- “Excluding Unidentified Software from the Scan” on page 589
- “Sorting Entries in the Table” on page 609
- “Filtering Entries in the Table” on page 609
- “Refreshing Entries in the Table” on page 610

3 Click OK.

Including Unidentified Software in a Scan

If you want unidentified software to be reported as a known software in subsequent scans, do the following:

1 Select the software entry in the table.

2 Click Software Dictionary located in the Add To pane.

The entry is automatically added to the Software Dictionary table.

Excluding Unidentified Software from the Scan

If you want unidentified software not to be reported in subsequent scans, do the following:

1. Select the software entry in the table.
2. Click Ignore Files located in the Add To pane.

   The entry is automatically added to the table in Software Scanning Filters - Files.

Report Multiple Software Versions

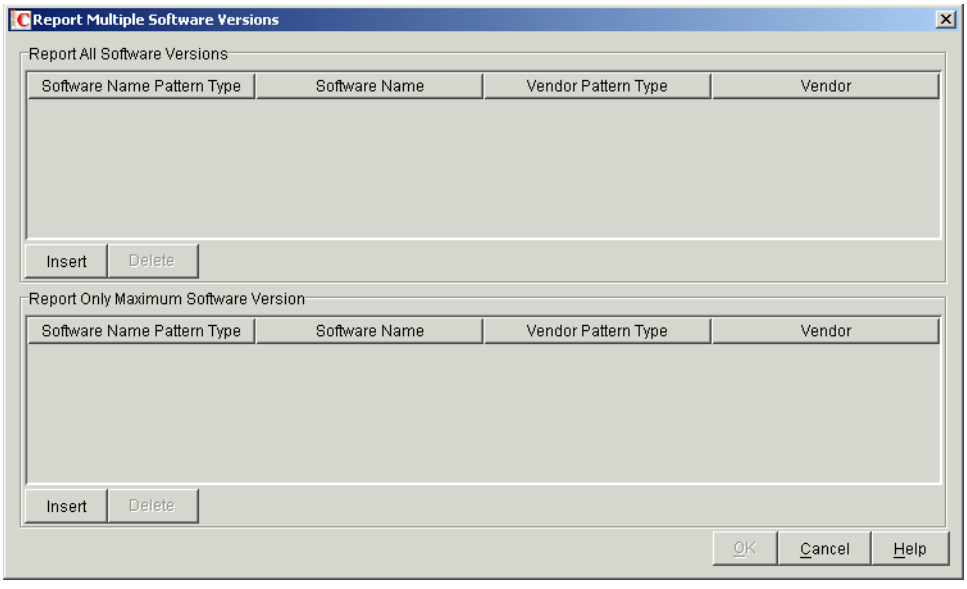
The “Report Multiple Software Versions” rule allows you to specify the software for which the Inventory scanner must report multiple versions installed on the inventoried server.

By default, the Inventory scanner scans for the highest version of the software installed on the inventoried server.

To configure this rule:

1. Click Edit Table.

   The Report Multiple Software Versions dialog box is displayed.



2. If you want the Inventory scanner to report all versions of the software installed on the inventoried servers, configure a rule in the Report All Software Versions table.
   - 2a In the Report All Software Versions table, click Insert to add a new row.
   - 2b In the Software Name Pattern Type drop-down list, select Expandable expression or Regular expression.
   - 2c Specify a software name.
   - 2d (Optional) In the Vendor Pattern Type drop-down list, select Expandable expression or Regular expression.
   - 2e (Optional) Specify a vendor name.

For example, if you want the Inventory scanner to report all versions of the Adobe Acrobat Reader installed on the inventoried server, configure the following settings in the table:

Software Name Pattern Type = Expandable expression

Software Name = Acrobat* Reader*

Vendor Pattern Type = Expandable expression

Vendor Name = Adobe*

If the inventoried server has Acrobat Reader versions 5.0 and 6.0 installed, the Inventory scanner reports both versions of Acrobat Reader (5.0 and 6.0).

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the table, select the entry and click delete. You can delete only the non-inherited entries.

3 If you want the Inventory scanner to report only the highest version of the software installed on the inventoried servers, configure a rule in the Report Only Maximum Software Version table.

3a In the Report Only Maximum Software Version table, click Insert to add a new row.

3b In the Software Name Pattern Type drop-down list, select Expandable expression or Regular expression.

3c Specify a software name.

3d (Optional) In the Vendor Pattern Type drop-down list, select Expandable expression or Regular expression.

3e (Optional) Specify a vendor name.

For example, if you want the Inventory scanner to report only the highest version of the Adobe Acrobat Reader installed on the inventoried server, configure the following settings in the table:

Software Name Pattern Type = Expandable expression

Software Name = Acrobat* Reader*

Vendor Pattern Type = Expandable expression

Vendor Name= Adobe*

If the inventoried server has Adobe Acrobat Reader versions 4.0 and 5.0 installed, then the Inventory scanner reports only Adobe Acrobat Reader 5.0.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the table, select the entry and click delete. You can delete only the non-inherited entries.

4 Click OK.

By default, the scanner reports only the highest version of the software installed. If a rule in Report All Software Versions conflicts with a rule in Report Only Maximum Software Version, then the rule in Report Only Maximum Software Version overrides the rule of Report All Software Versions.

For example, if you want the Inventory scanner to report all versions of Microsoft software except for Microsoft Office, and also report only the highest version of Microsoft Office installed, configure the following filters as shown below:

- **Report All Software Versions:** Configure the following settings:

  Software Name Pattern Type = Expandable expression
  Software Name = *
  Vendor Pattern Type = Expandable expression
  Vendor Name= Microsoft*

- **Report Only Maximum Version:** Configure the following settings:

  Software Name Pattern Type = Expandable expression
  Software Name = *office*
  Vendor Pattern Type = Expandable expression
  Vendor Name= Microsoft*

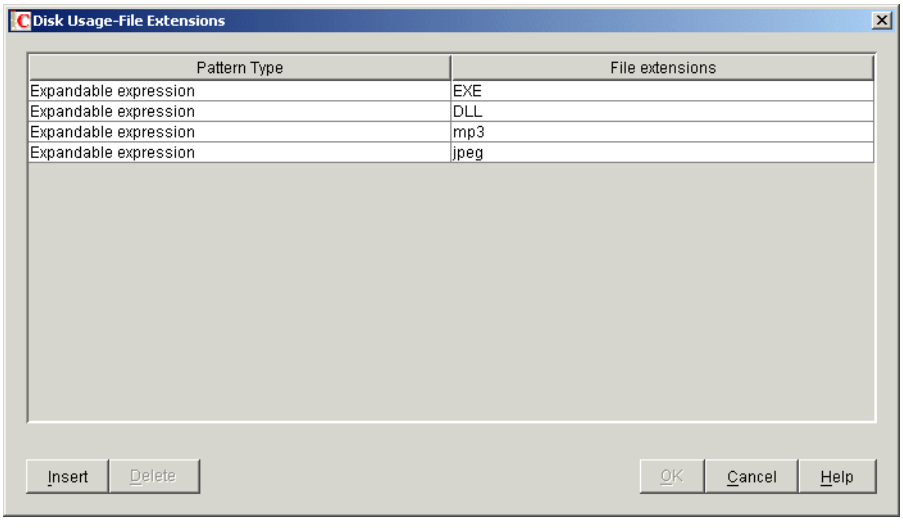
Report Disk Space Used by File Extensions

The “Report Disk Space Used by File Extensions” rule allows you to specify the file extension of the files whose total disk usage you want to scan.

To configure this rule:

1. Click Edit Table.

   The Disk Usage - File Extensions table is displayed.



2. Click Insert to add a new row.
3. In the Pattern Type drop-down list, select Expandable expression or Regular expression.
4. Specify a file extension.
5. Click OK.

For example, if you want the Inventory scanner to scan for disk usage of all files with extension ".pif", configure the following settings in the Disk Usage - File Extensions table:

Pattern Type = Expandable expression

File Extension = pif

The Inventory scanner scans and stores only the total disk usage for all files with extension ".pif" in the Inventory database.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Disk Usage - File Extensions table, select the entry and click Delete. You can delete only the non-inherited entries.

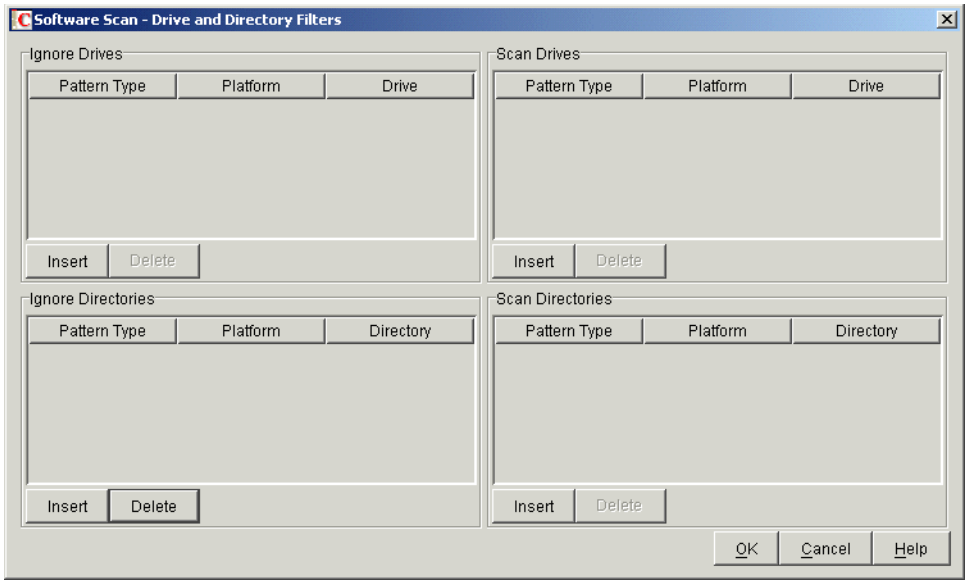
Software Scanning Filters - Drives and Directories

The "Drives and Directories" filter the allows you to control the scanning for software files located in specified drives and directories.

To configure this filter:

1 Click Edit Table.

The Software Scan - Drive and Directory Filters dialog box is displayed.



2 Configure the following filters:

- "Ignore Drives" on page 593
- "Scan Drives" on page 594
- "Ignore Directories" on page 594
- "Scan Directories" on page 595

By default, the Inventory scanner scans all directories on the inventoried servers. If you have configured a rule that ignores all directories during a scan by using the Ignore Directories filter, but now want to include a specific directory in a scan, you can identify the specific directory using the Scan Directories filter. The settings of the Scan Directories filter overrides the settings of the Ignore Directories and Ignore Drives filters.

For example, if you want the Inventory scanner to ignore all files and directories in C: except for the c:\program files directory on Windows inventoried servers, configure the following filters as shown below:

- **Ignore Drives:** Configure the following settings:

  Pattern Type = System expandable expression
  Platform = Windows
  Drive = C

- **Scan Directories:** Configure the following settings:

  Pattern Type = System expandable expression
  Platform = Windows
  Drive = c:\program files

3. Click OK.

Ignore Drives

The “Ignore Drives” filter allows you to specify the drives that should not be scanned for on the inventoried servers.

By default, the Inventory scanner scans all drives.

To configure this filter:

1. In the Ignore Drives table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.

4. Specify a drive name.

For example, if you want the Inventory scanner not to scan the files in C drive on all the Windows inventoried servers, configure the following settings in the Ignore Drives table:

Pattern Type = System expandable expression
Platform = Windows
Drive = C

The Inventory scanner does not scan the files in the C drive.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Ignore Drives table, select the entry and click Delete. You can delete only the non-inherited entries.

Scan Drives

The “Scan Drives” filter allows you to specify the drives that should be scanned for at the inventoried servers.

To configure this filter:

1. In the Scan Drives table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a drive name.

For example, if you want the Inventory scanner to scan for files in the C drive on all the Windows inventoried servers, configure the following settings in the Scan Drives table:

Pattern Type = System expandable expression
Platform = Windows
Drive = C

You must also configure the following settings in the Ignore Drives table:

Pattern Type = System expandable expression
Platform = Windows
Drive = *

The Inventory scanner scans only the files in the C drive for the software information.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Scan Drives table, select the entry and click Delete. You can delete only the non-inherited entries.

IMPORTANT: By default, the Inventory scanner scans all drives on the inventoried servers. If you have configured all drives to be ignored during a scan by using the Ignore Drives filter, but now want to include a specific drive in a scan, you can identify the specific drive using the Scan Drives filter. The settings of the Scan Drives filter override the settings of the Ignore Drives filter.

Ignore Directories

The “Ignore Directories” filter allows you to specify the directories that should not be scanned for at the inventoried servers.

By default, the Inventory scanner scans all directories.

To configure this filter:

1. In the Ignore Directories table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a directory name.

For example, if you do not want the Inventory scanner to scan the files in the c:\program files directory on all the Windows inventoried servers, configure the following settings in the Ignore Directories table:

Pattern Type = System expandable expression

Platform = Windows

Directory= C:\Program Files

The Inventory scanner does not scan for the files in c:\program files.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Ignore Directories table, select the entry and click Delete. You can delete only the non-inherited entries.

Scan Directories

The “Scan Directories” filter allows you to specify the directories that should be scanned for at the inventoried servers.

To configure this filter:

1. In the Scan Directories table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a directory name.

For example, if you want the Inventory scanner to scan for files in the c:\program files directory on all the Windows inventoried servers, configure the following settings in the Scan Directories table:

Pattern Type = System expandable expression

Platform = Windows

Directory = C:\Program Files

You must also configure the following settings in the Ignore Directories table:

Pattern Type = System expandable expression
Platform = Windows
Directory = *

The Inventory scanner scans only the files in c:\program files for software information.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Scan Directories table, select the entry and click Delete. You can delete only the non-inherited entries.

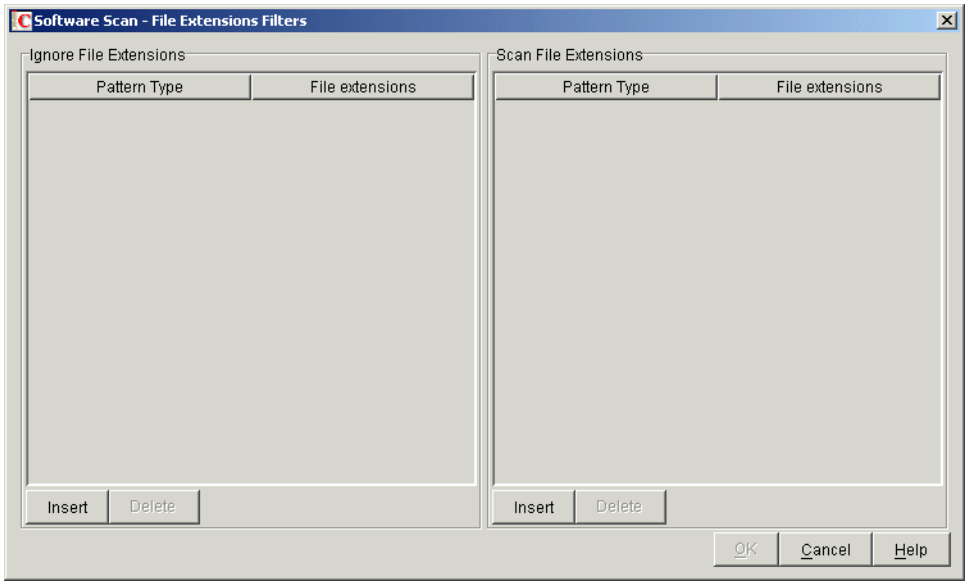
Software Scanning Filters - File Extensions

The “File Extensions” filter allows you to control scanning for software files with a specified extension.

To configure this filter:

1 Click Edit Table.

The Software Scan - File Extensions Filters dialog box is displayed.



2 Configure the following filters:

- “Ignore File Extensions” on page 597
- “Scan File Extensions” on page 597

3 Click OK.

Ignore File Extensions

The “Ignore File Extensions” filter allows you to specify the file extensions that should not be scanned for at the inventoried servers.

To configure this filter:

1. In the Ignore File Extensions table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select Expandable expression or Regular expression.
3. Specify a file extension.

For example, if you do not want the Inventory scanner to scan for files whose extension begins with “.ex,” configure the following settings in the Ignore File Extensions table:

Pattern Type = Expandable expression
File Extension = ex*

The Inventory scanner does not scan for the files whose extension begin with “.ex.” For example, .ex1, .ex2, .exe, and exec.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Ignore File Extensions table, select the entry and click Delete. You can delete only the non-inherited entries.

Scan File Extensions

The “Scan File Extensions” filter allows you to specify the file extensions that should be scanned for at the inventoried servers.

If you have excluded file extensions from scanning by using the Ignore File Extensions filter, but now want to include a specific file extension in the scan, you can identify the specific file extension using the Scan File Extensions filter. The settings of the Scan File Extensions filter override the settings of the Ignore File Extensions filter.

To configure this filter:

1. In the Scan File Extensions table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select Expandable expression or Regular expression.
3. Specify a file extension.

For example, if you want the Inventory scanner to scan for all files with a “.exe” extension, configure the following settings in the Scan File Extension table:

Pattern Type = Regular expression
File Extension = [exe|EXE]

The Inventory scanner scans and stores only the files with extension “.exe” in the Inventory database.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Scan File Extensions table, select the entry and click Delete. You can delete only the non-inherited entries.

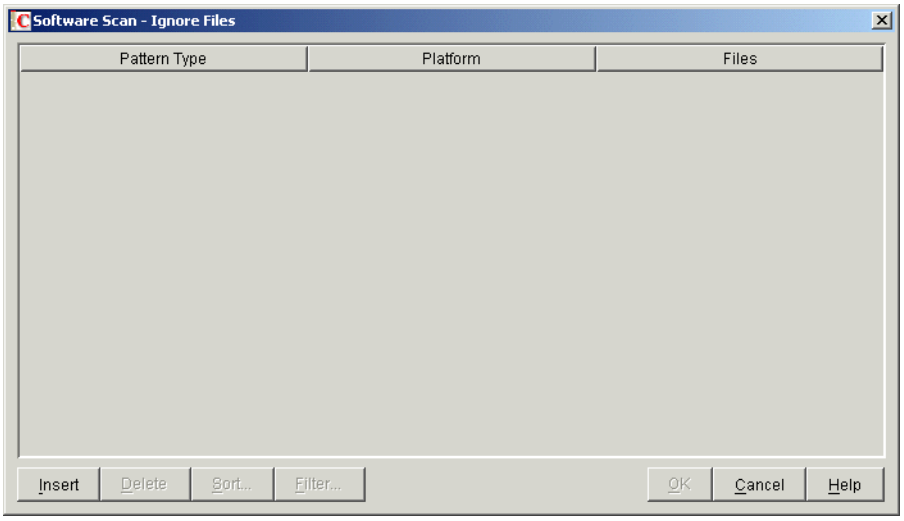
Software Scanning Filters - Files

The “Files” filter allows you to configure software to be excluded during the Inventory scan.

To configure this filter:

1. Click Edit Table.

   The Software Scan - Ignore Files table is displayed.



2. Click Insert to add a new row.
3. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
4. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column is automatically changed to “Any.” You cannot change the value.
5. Specify a filename.
6. Click OK.

For example, if you want the Inventory scanner to scan notepad.exe on all the Windows inventoried servers, configure the following settings:

Platform = Windows
Pattern Type = System expandable expression
Files=notepad.exe

This table also displays files that are added from the Manage Unidentified Software table.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

You can also perform the following operations in the Software Scan - File Filters table:

- Deleting only the non-inherited entries.
- Sorting Entries in the Table.
- Filtering Entries in the Table.
- Refreshing Entries in the Table.

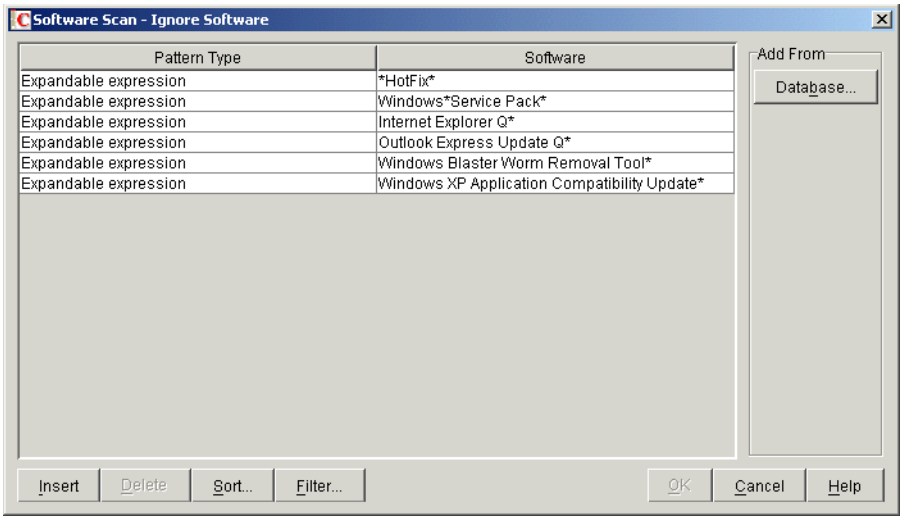
Software Scanning Filters - Software

The “Software” filter allows you to configure a software that is not to be reported during the Inventory scan.

To configure this filter:

1 Click Edit Table.

The Software Scan - Ignore Software table is displayed.



2 You can add entries to the Ignore Software table either manually or automatically.

Manually Adding Entries to the Table

1. Click Insert to add a new row.
2. In the Pattern Type drop-down list, select Expandable expression or Regular expression.
3. Specify a software name.
4. Click OK.

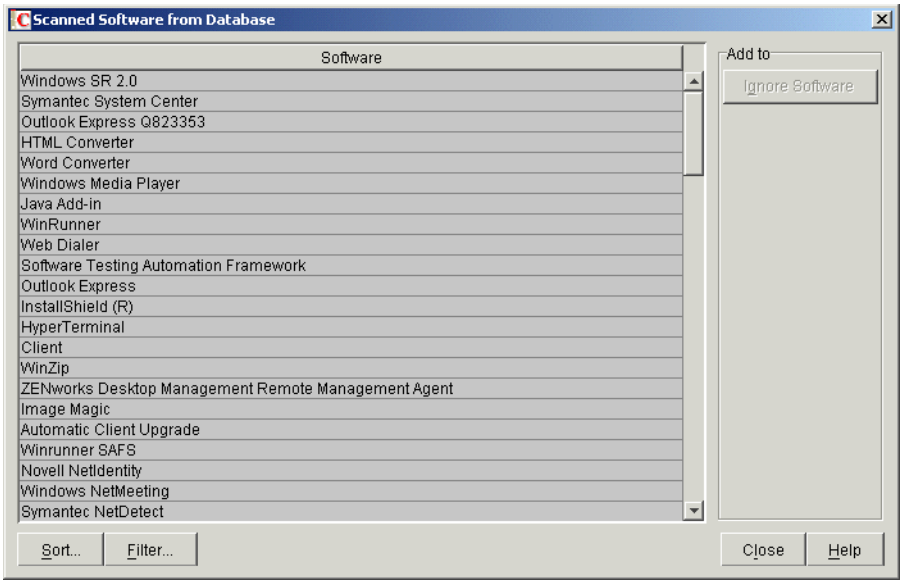
For example, if you do not want the Inventory scanner to scan for the Adobe products, configure the following settings:

Pattern Type= Expandable expression
Software = Adobe*

The Inventory scanner does not report the software that has names beginning with “Adobe.”

Automatically Adding Entries to the Table

1. Click Database located in the Add From pane.
   The Scanned Software from Database dialog box is displayed.



2. Select the software that you want to add to the Ignore Software table.
3. Click the Ignore Software button located in the Add to pane.
4. Click Close.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

You can also perform the following operations in the Software Scan - Ignore Software table:

- Deleting only the non-inherited entries.
- Sorting Entries in the Table.
- Filtering Entries in the Table.
- Refreshing Entries in the Table.

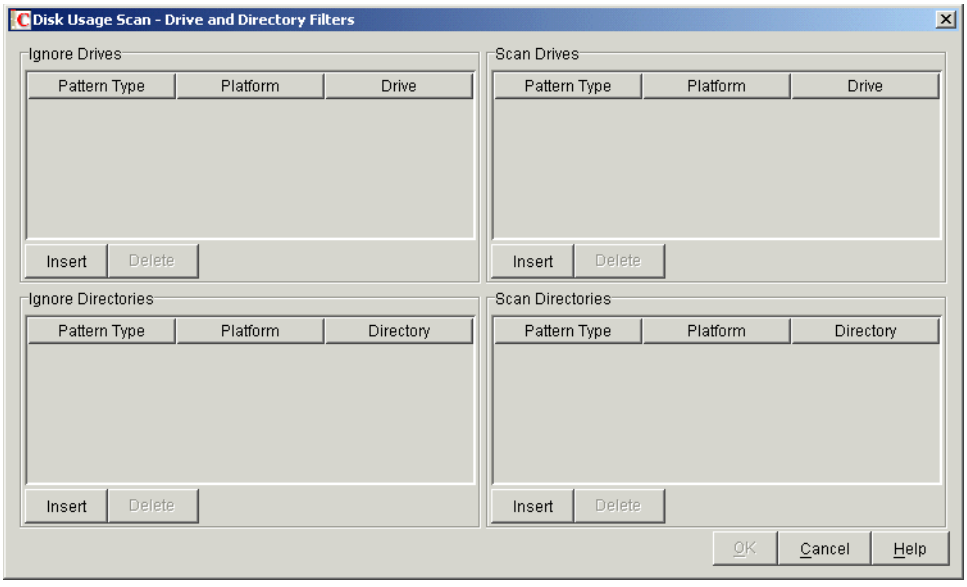
Disk Usage Scanning Filters - Drives and Directories

The “Drives and Directories” filter the allows you to configure drives and directories to be included or excluded during the Inventory disk usage scanning.

To configure this filter:

1 Click Edit Table.

The Disk Usage Scan - Drive and Directory Filters dialog box is displayed.



2 Configure the following filters:

- “Ignore Drives” on page 602
- “Scan Drives” on page 602
- “Ignore Directories” on page 603
- “Scan Directories” on page 604

By default, the Inventory scanner scans the disk usage of all directories on the inventoried servers. If you have configured all directories to be ignored during a disk usage scan using the Ignore Directories filter, but now want to include a specific directory in scan, identify the specific directory in the Scan Directories filter. The settings of the Scan Directories filter override the settings of the Ignore Directories and Ignore Drives filters.

For example, if you want the Inventory scanner to ignore the disk usage of all files and directories in C: except for the c:\program files directory on Windows inventoried servers, configure the following filters as shown below:

- **Ignore Drives:** Configure the following settings:

  Pattern Type = System expandable expression
  Platform = Windows
  Drive = C

- **Scan Directories:** Configure the following settings:

  Pattern Type = System expandable expression
  Platform= Windows
  Drive=c:\program files

3 Click OK.

Ignore Drives

The “Ignore Drives” filter allows you to specify the drives that should not be scanned for disk usage at the inventoried servers.

By default, the Inventory scanner scans all drives.

To configure the “Ignore Drives” filter:

1. In the Ignore Drives table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a drive name.

For example, if you want the Inventory scanner not to scan for the disk usage of C drive on all the Windows* inventoried servers, configure the following settings in the Ignore Drives table:

Pattern Type = System expandable expression
Platform = Windows
Drive = C

The Inventory scanner does not scan the disk usage of files on the C drive.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Ignore Drives table, select the entry and click Delete. You can delete only the non-inherited entries.

Scan Drives

The “Scan Drives” filter allows you to specify the drives whose disk usage should be scanned for at the inventoried servers.

To configure the “Scan Drives” filter:

1. In the Scan Drives table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a drive name.

For example, if you want the Inventory scanner to scan for the disk usage of C drive on all the Windows inventoried servers, configure the following settings in the Scan Drives table:

Pattern Type = System expandable expression

Platform = Windows

Drive = C

The Inventory scanner scans and stores the disk usage of the files in the C drive into the Inventory database.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Scan Drives table, select the entry and click Delete. You can delete only the non-inherited entries.

IMPORTANT: By default, the Inventory scanner scans the disk usage of all drives on the inventoried servers. If you have configured all drives to be ignored during a disk usage scan using the Ignore Drives filter, but now want to include a specific drive in the scan, identify the specific drive in the Scan Drives filter. The settings of the Scan Drives filter override the settings of the Ignore Drives filter.

Ignore Directories

The “Ignore Directories” filter allows you to specify the directories whose disk usage should not be scanned for at the inventoried servers.

By default, the Inventory scanner scans all directories.

To configure the “Ignore Directories” filter:

1. In the Ignore Directories table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a directory name.

For example, if you want the Inventory scanner not to scan for the disk usage of the c:\program files directory on all the Windows inventoried servers, configure the following settings in the Ignore Directories table:

Pattern Type = System expandable expression

Platform = Windows

Directory= c:\program files

The Inventory scanner does not scan for the disk usage of c:\program files.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Ignore Directories table, select the entry and click Delete. You can delete only the non-inherited entries.

Scan Directories

The “Scan Directories” filter allows you to specify the directories whose disk usage should be scanned for at the inventoried servers.

To configure the “Scan Directories” filter:

1. In the Scan Directories table, click Insert to add a new row.
2. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
3. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried servers.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
4. Specify a directory name.

For example, if you want the Inventory scanner to scan for disk usage of the c:\program files directory on all the Windows inventoried servers, configure the following settings in the Scan Directories table:

Pattern Type = System expandable expression

Platform = Windows

Directory = c:\program files

The Inventory scanner scans and stores only disk usage of files in c:\program files into the Inventory database.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

To delete an entry from the Scan Directories table, select the entry and click Delete. You can delete only the non-inherited entries.

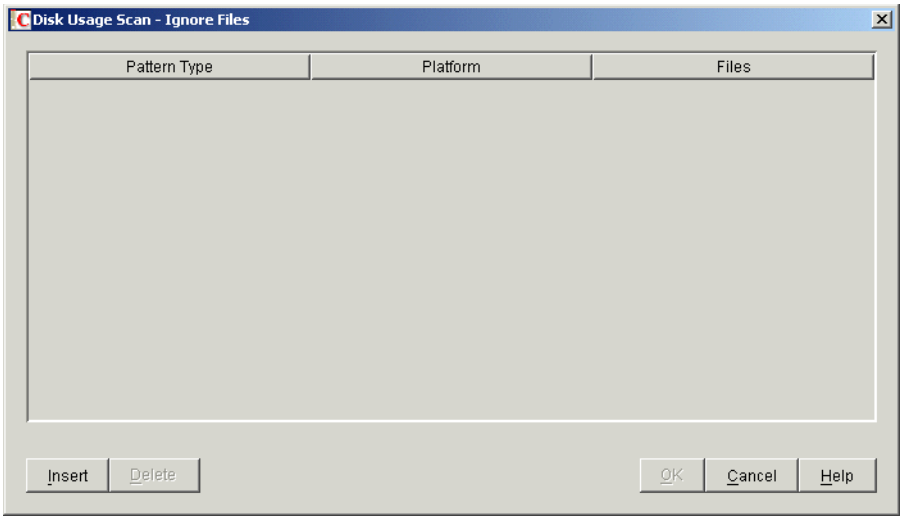
Disk Usage Scanning Filters - Files

The “Files” filter allows you to configure files to be excluded during the Inventory disk usage scanning.

To configure this filter:

1. Click Edit Table

   The Disk Usage Scan - Ignore Files dialog box is displayed.



2. Click Insert to add a new row.
3. In the Pattern Type drop-down list, select System expandable expression, Expandable expression, or Regular expression.
4. (Conditional) If you select System expandable expression as the pattern type, then select NetWare or Windows in the Platform drop-down list, depending on the operating system of the inventoried serves.

   IMPORTANT: If you select Expandable expression or Regular expression as the pattern type, the corresponding value in the Platform column automatically changes to “Any.” You cannot change the value.
5. Specify a file.
6. Click OK.

For example, if you want the Inventory scanner to scan for disk usage of all files with extension “.exe” except msoffice.exe, configure the following rules as shown below:

- **Disk Usage Scan - Ignore Files:** Configure the following settings:

  Pattern Type = Expandable expression

  Files = msoffice.exe
- **Report Disk Space used by file extensions:** Configure the following settings:

  Pattern Type = Expandable expression

  Files = exe

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

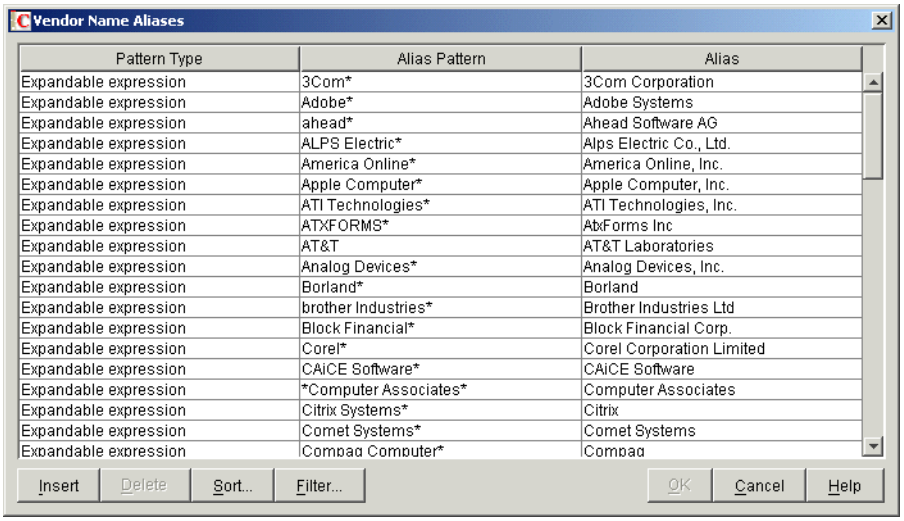
To delete an entry from the table, select the entry and click Delete. You can delete only the non-inherited entries.

Vendor Name Aliases

The “Vendor Name Aliases” rule allows you to configure aliases for vendor names.

1 Click Edit Table.

The Vendor Name Aliases table is displayed.



2 Click Insert to add a new row.

3 In the Pattern Type drop-down list, select Expandable expression or Regular expression.

4 Specify an alias pattern.

5 Specify an alias.

6 Click OK.

For example, if you want the Inventory scanner to report all instances of the vendor name beginning with “Microsoft” as “Microsoft Corporation” in the Inventory database, configure the following settings:

Pattern Type = Expandable expression

Alias Pattern = Microsoft*

Alias = Microsoft Corporation

If the Inventory scanner reports Microsoft, Microsoft Inc., or Microsoft Inc. Corporation vendor names during the scan, then the name of the vendor beginning with “Microsoft” is stored as “Microsoft Corporation” in the Inventory database.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

You can also perform the following operations in the Vendor Name Aliases table:

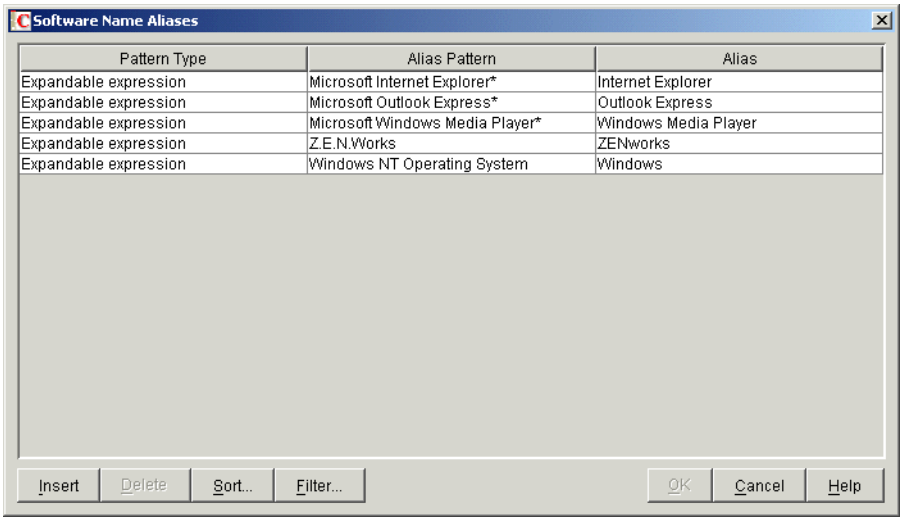
- Deleting only the non-inherited entries.
- Sorting Entries in the Table.
- Filtering Entries in the Table.
- Refreshing Entries in the Table.

Software Name Aliases

The “Software Name Aliases” rule allows you to configure aliases for software names.

1 Click Edit Table.

The Software Name Aliases table is displayed.



2 Click Insert to add a new row.

3 In the Pattern Type drop-down list, select Expandable expression or Regular expression.

4 Specify an alias pattern.

5 Specify an alias.

6 Click OK.

For example, if you want the Inventory scanner to report all instances of the product name “WinZip” as “WinZip Application” in the Inventory database, configure the following settings:

Pattern Type = Expandable expression
Alias Pattern = WinZip
Alias = WinZip Application

If the Inventory scanner scans the WinZip, WinZip Executables, or WinZip Applications product names, then the name of the software that exactly matches “WinZip” is stored as “WinZip Application” in the Inventory database. The remaining software names are reported as scanned.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

You can also perform the following operations in the Software Name Aliases table:

- Deleting only the non-inherited entries.
- Sorting Entries in the Table.
- Filtering Entries in the Table.
- Refreshing Entries in the Table.

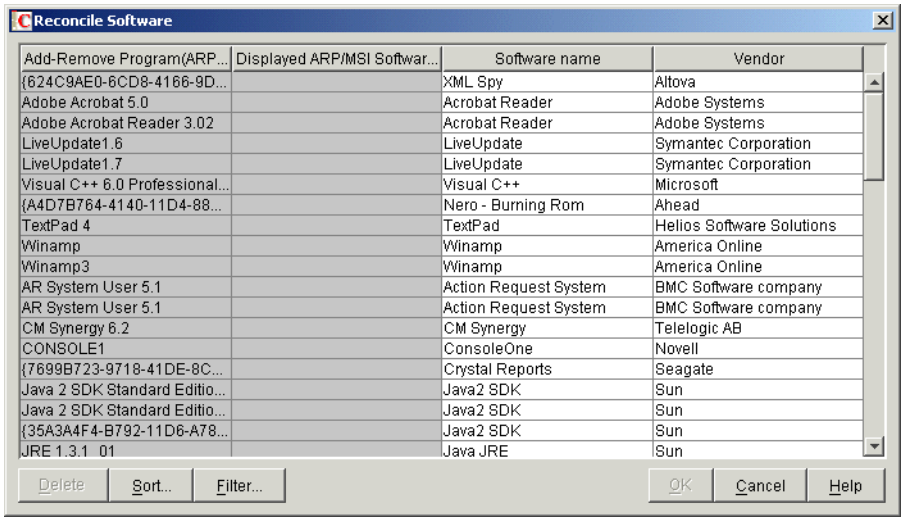
Reconcile Software

The “Reconcile Software” rule allows you to associate the software identified through Add/Remove Programs or the MSI, with an appropriate software and vendor identified and configured through the ZENworks software dictionary. The association might be necessary because the software entries in Add/Remove Programs or the MSI might not use the same software name and vendor as configured in the ZENworks software dictionary.

To configure the rule:

1 Click Edit Table.

The Reconcile Software table is displayed.



By default, the Reconcile Software table displays pre-defined mapping of software in the Add/Remove Programs or MSI with the software configured in the software dictionary. It also displays the Add/Remove Programs or MSI software identified during the last scan for which you can configure software and vendor names. This table has the following columns.

- The Add-Remove Program (ARP) key.

  You cannot edit the values of this attribute.

- The ARP /MSI name as displayed either in Add/Remove Programs or in the MSI.

  The Displayed ARP/MSI Software name shows the software identified through Add/Remove Programs or the MSI and stored in the Inventory database.

  You cannot edit the values of this attribute.

- The software name associated with its corresponding Add/Remove Programs or MSI name.

- The vendor name associated with its corresponding Add/Remove Programs or MSI name.

IMPORTANT: The table might contain inherited rules. You can edit or delete these rules only in the software dictionary of the inventory server configured in the Dictionary Update policy. These rules are displayed in dark gray color if you are accessing the software dictionary of ZENworks 6.5 SP1 or later Inventory server.

2 To reconcile software, do the following in this table:

- Specify software and vendor names for the software identified through Add/Remove Programs or the MSI but not yet been associated.
- (Optional) Change the software and vendor names for the software that has already been configured in the software dictionary.

You can also perform the following operations in the Reconcile Software table:

- Deleting only the non-inherited entries.
- Sorting Entries in the Table.
- Filtering Entries in the Table.
- Refreshing Entries in the Table.

Sorting Entries in the Table

You can sort the entries in the table by one, two, or three columns.

1 Click Sort.

The Sort dialog box is displayed.

2 In the Sort by drop-down list, select the table column by which you want to sort the entries.

3 Select Ascending or Descending.

4 (Optional) To sort by either two or three columns, configure the Then by drop-down lists and select Ascending or Descending.

5 Click OK

Filtering Entries in the Table

1 Click Filter.

The Filter dialog box is displayed.

2 In the Filter dialog box, do the following to create a query:

2a Select an attribute.

2b Select an operator. The operators displayed depend on the attribute you select in Step 2a.

2c Type a value.

2d (Optional) To create an advanced query, select one of the following logical operators and define the query:

Logical Operator	Functionality
AND	Creates a new row. The filter displays items that match the conditions in each row joined by AND.
OR	Creates a new row. The filter displays items that match the conditions in either row joined by OR.
New Row	Creates a new row to form a new query.
Delete Row	Deletes the row from the filter.

Logical Operator	Functionality
End	Closes the query expression. If you select End in a row that is followed by other rows, the subsequent rows and groups are deleted.

3 Click OK.

After applying the filter, the table list displays only the resulting entries. To clear the filter:

1 Click Filter.

The Filter dialog box is displayed.

2 Click Clear, then click OK.

Refreshing Entries in the Table

Use the Refresh option if you want to reapply the sort or filter operations. To refresh entries in the table, either click Refresh or press F5.

IMPORTANT: The Refresh button is displayed only when you apply sort or filter operations to the table.

Disabling File Scan

You can disable the software scanning for all software except for the following:

- Software registered in the Add/Remove Programs dialog box
- Software installed through MSI
- Software scanned by default such as Microsoft Windows, Internet Explorer, Outlook, MediaPlayer, ZENworks, Novell client, Microsoft Office and the set of Antivirus programs

To disable the file scanning, review the following sections:

- “Disabling File Scan in ZENworks 6.5” on page 610
- “Disabling File Scan in ZENworks 6.5 SP1 or Later” on page 611

Disabling File Scan in ZENworks 6.5

1 In ConsoleOne, right-click the Inventory Service Object, click Properties.

2 Click the Software Inventory Configuration tab, then click the Software Dictionary sub-option.

3 Click the Edit Table button of the “Scan as Unidentified Software” rule.

4 By default, the table has an entry with the EXE file extension. Delete the entry.

5 Click OK.

6 Click Apply.

7 In the Software Inventory Configuration tab, click the Disk Usage Scanning sub-option.

8 Click the Edit Table button of the “Include Local File Extensions” rule.

9 By default, the table has entries with the EXE, DLL, MP3, and JPEG file extensions. Delete all entries.

10 Click OK.

11 Click Apply, then click Close.

Disabling File Scan in ZENworks 6.5 SP1 or Later

1 In ConsoleOne, right-click the Inventory Service Object, click Properties.

2 Click the Software Inventory Configuration tab.

The Software Configuration page is displayed by default.

3 Select the “Ignore Default File-Software Mapping Rules” check box.

4 Click the Edit Table button of the “Report Files with These File Extensions As Unidentified Software” rule.

5 By default, the table has an entry with the EXE file extension. Delete the entry.

6 Click OK.

7 Click Apply, then click Close.

Base-lining the ZENworks 6.5 Software Dictionary Deployment

When you deploy the software dictionary for the first time, the default dictionary settings will be effective and the Inventory scanner will report the following information:

- Unidentified software
- Multiple instances of software installed on the same inventoried server
- Redundant drives and directories that do not contain software

This scan results in storing huge amount of irrelevant information in the Inventory database. It also degrades the performance of Storer and all Inventory ConsoleOne utilities such as Query, Reporting, etc.

To avoid these problems, we recommend you to fine tune the software dictionary according to your requirements before deploying it in your enterprise. Do the following to fine tune the software dictionary:

1 Deploy a small representative set of inventoried servers in the test lab.

NOTE: This representative set should typically represent all sections or departments of your enterprise that you want to collect inventory for.

2 Attach these inventoried servers to a Standalone Inventory Server, which is also deployed in the test lab.

3 In addition to defaults that are listed in the table of the “Scan as Unidentified Software” on page 571 rule, you may want to scan for additional file extensions and report them as identified software. These could be extensions of application files like DLLs, etc.

4 Schedule the scan and wait till the inventory information is stored into the Inventory database.

5 Re-configure the software dictionary based on the inventory information that is available in the database to resolve the above discussed problems:

- **UnIdentified Software:** Based on the Scan as Unidentified Software settings, all the information related to the unidentified software can be viewed in the “Add Unidentified Software to Dictionary” on page 570 table.

The result will contain the following:

- Applications that are not yet identified by the software dictionary.
- Application files that are part of already identified by the software dictionary.
- Application files that might be redundant. For example, Operating System files, DOS files, etc.

You must first add these files to the software dictionary using the “Software Dictionary” button (located in the “Add Unidentified Software to Dictionary” on page 570 table). And then remove these settings from the “Scan as Unidentified Software” on page 571 rule.

NOTE: If you want to add these files to be reported as Unidentified later, you must again add these files to the “Scan as Unidentified Software” on page 571 rule.

The effectiveness of this exercise is based on the following assumptions:

- The representative set should not be different from the sections or departments of your enterprise; otherwise it would amount for large number of un-identified software being scanned and reported.
- The inventoried servers in the enterprise are largely controlled by the enterprise administrator, who installs and copies the non-standard applications.

- **Multiple instances of Software on the same inventoried server:** For an inventoried server, the same software can be reported twice if one entry is reported from the Add Remove Program scanning or the MSI scanning, and the other is reported based on the software dictionary configuration. The “Edit Add-Remove Software” on page 580 rule contains default configurations to merge these two entries but this may not be complete. In order to resolve this problem, you must manually configure the Edit Add-Remove Software rule.
- **Redundant drives and directories that do not contain software:** Configure the rules in Software Scanning page and the Disk Usage Scanning pages of the software dictionary to eliminate these drives and directories from scan. For more information about the software dictionary rules, see “Configuring the ZENworks 6.5 SP1 or Later Software Dictionary Rules” on page 580.

6 Re-scan all the inventoried servers.

7 After the inventory information is stored in the Inventory database, you could notice that all the entries that you marked for dictionary during the earlier scan would be scanned and reported as a software.

8 Repeat Step 3 through Step 7 till you fine tune the dictionary according to your requirements.

Base-lining the ZENworks 6.5 SP1 or Later Software Dictionary Deployment

When you deploy the software dictionary for the first time, the default dictionary settings will be effective and the Inventory scanner will report the following information:

- Unidentified software
- Multiple instances of software installed on the same inventoried server
- Redundant drives and directories that do not contain software

This scan results in storing huge amount of irrelevant information in the Inventory database. It also degrades the performance of Storer and all Inventory ConsoleOne utilities such as Query, Reporting, etc.

To avoid these problems, we recommend you to fine tune the software dictionary according to your requirements before deploying it in your enterprise. Do the following to fine tune the software dictionary:

1. Deploy a small representative set of inventoried servers in the test lab.

   NOTE: This representative set should typically represent all sections or departments of your enterprise that you want to collect inventory for.

2. Attach these inventoried servers to a Standalone Inventory Server, which is also deployed in the test lab.
3. In addition to defaults that are listed in the table of the Report Files with These File Extensions As Unidentified Software rule, you may want to scan for additional file extensions and report them as identified software. These could be extensions of application files like DLLs, etc.
4. Schedule the scan and wait till the inventory information is stored into the Inventory database.
5. Re-configure the software dictionary based on the inventory information that is available in the database to resolve the above discussed problems:

   - **UnIdentified Software:** Based on the Report Files with These File Extensions As Unidentified Software settings, all the information related to the unidentified software can be viewed in the Manage Unidentified Software table.

     The result will contain the following:

     - Applications that are not yet identified by the software dictionary.
     - Application files that are part of already identified by the software dictionary.
     - Application files that might be redundant. For example, Operating System files, DOS files, etc.

     Perform the following tasks in the Manage Unidentified Software table:

     - Add the applications that are not yet identified by the software dictionary to the Software Dictionary table using the Software Dictionary button located in the Add To pane.
     - Add the application files that are part of already identified by the software dictionary and application files that might be redundant to the Software Scanning Filters - Files table using the Ignore Files button located in the Add To pane.

     The effectiveness of this exercise is based on the following assumptions:

     - The representative set should not be different from the sections or departments of your enterprise; otherwise it would amount for large number of un-identified software being scanned and reported.
     - The inventoried servers in the enterprise are largely controlled by the enterprise administrator, who installs and copies the non-standard applications.

   - **Multiple instances of Software on the same inventoried server:** For an inventoried server, the same software can be reported twice if one entry is reported from the Add Remove Program scanning or the MSI scanning, and the other is reported based on the software dictionary configuration. The “Reconcile Software” on page 608 rule contains default configurations to merge these two entries but this may not be complete. In order to resolve this problem, you must manually configure the Edit Add-Remove Software rule.

- **Redundant drives and directories that do not contain software:** Configure the rules in Software Scanning page and the Disk Usage Scanning pages of the software dictionary to eliminate these drives and directories from scan. For more information about the software dictionary rules, see Step 3 on page 581.

6 Re-scan all the inventoried servers.

7 After the inventory information is stored in the Inventory database, you could notice that all the entries that you marked for dictionary during the earlier scan would be scanned and reported as a software.

8 Repeat Step 3 through Step 7 till you fine tune the dictionary according to your requirements.

Viewing Software Inventory Information in the Inventory Summary

To view the software inventory information of an inventoried server in the Inventory Summary:

1 In ConsoleOne, configure the Inventory database. For more information on how to configure the database, see "Configuring the Inventory Database" on page 619

2 Right-click an inventoried server, click Actions, then click Inventory.

3 In the Summary dialog box, click Inventory Information > Hardware/Software Inventory > Software > Application Vendors to view the software inventory information.

On choosing Application Vendor, a list of Software Group and Software of the vendor are displayed. Software Group can contain software patch and representative file information of the group. Software can contain software patch and representative file information of the product.

For more information, see "Viewing the Inventory Summary of an Inventoried Server" on page 620

Generating Software Inventory Reports

You can now generate the following Software Inventory reports:

- Add-Remove Programs by Machine
- Anti-Virus Signature Files by Machine
- Anti-Virus Signature Machine Count
- Disk Usage by Machine
- Exception List by Machine
- Installed NetWare Software by Machine
- Internet Explorer Installation Count
- Internet Explorer Patches by Machine
- Internet Explorer by Machine
- MSI Products by Machine
- Microsoft Office Components by Machine
- Microsoft Office Installation Count
- Microsoft Office by Machine
- Novell Client Components by Machine

- Novell Installation Count
- Novell Client by Machine
- Novell ZENworks Desktop Management Installed Agent Components by Machine
- Novell ZENworks Desktop Management Installed Server Components by Machine
- Novell ZENworks Handheld Management Installed Components by Machine
- Novell ZENworks Installed Components by Machine
- Novell ZENworks Installed Suites by Machine
- Novell ZENworks Server Management Installed Agent Components by Machine
- Novell ZENworks Server Management Installed Server Components by Machine
- Outlook Express Installation Count
- Outlook Express by Machine
- Software Dictionary Application Files by Machine
- Software Dictionary Applications by Machine
- Software Dictionary Versions Machine Count
- Software Dictionary Versions by Machine
- Software Installation Count
- Software Installations
- Software by Machine
- System Software Inventory Report
- Windows Components by Machine
- Windows Installation Count
- Windows Media Player Count
- Windows Media Player Patches by Machine
- Windows Media Player by Machine
- Windows Operating System by Machine
- Windows Security Patches by Machine

For more information about each report, see “Types of Inventory Reports” on page 636.

Customizing the Software Inventory Information To Be Scanned For ZENworks for Servers 3.*X* Inventoried Servers

Refer to the ZENworks for Servers 3.0.2 Documentation Web site (http://www.novell.com/documentation/zfs302/index.html) to know how to customize the software inventory information for the ZENworks for Servers 3.*x* inventoried servers.

Removing the Redundant Inventoried Servers from the Inventory Database

You can remove the unwanted, redundant, or obsolete inventoried servers from the Inventory database using the Inventory Removal service.

The Inventory Removal service is a manual service that runs on the Inventory server. The service removes the inventoried servers from the Inventory database using the inventoryremovallist.txt file. The inventoryremovallist.txt file contains a list of inventoried servers that must be removed from the Inventory database.

IMPORTANT: You can run the Inventory Removal service on the Intermediate Server only if the Intermediate Server has either inventoried servers or database attached to it.

To remove the inventoried servers from the Inventory database:

1. Using a text editor, create a file with the name inventoryremovallist.txt with the following contents:

   ```
   ;                    Enter comments, if any

   DN of the inventoried server (as stored in the Inventory database) to be
   removed from the Inventory database
   DN of the inventoried server (as stored in the Inventory database) to be
   removed from the Inventory database
   ....
   DN of the inventoried server (as stored in the Inventory database) to be
   removed from the Inventory database
   ```

   A sample inventoryremovallist.txt file is as follows:

   ```
   CN=INT-SERVER-NDS.OU=Leaf.O=XYZ.T=XYZ-TREEzen-server.xyz.com
   CN=ROOT-SERVER-NDS.O=XYZ.T=XYZ-TREE
   ```

   To generate the list of inventoried servers that must be removed you can either perform a query on a selected criteria or manually enter the names of the inventoried servers. For more information on Query, see “Viewing Inventory Information of Inventoried Servers by Querying the Database” on page 632.

2. Copy the inventoryremovallist.txt file to the *ZENworks_installation_path*\zenworks\inv\server\wminv\properties directory.

3. In the *ZENworks_installation_path*\zenworks\inv\server\wminv\properties\ inventoryremoval.properties file, ensure that the value of FilePath is the location of inventoryremovallist.txt (specified in Step 2).

   NOTE: Ensure that the path separator is a forward slash (/) and not a backslash (\ ).

4. At the server console prompt, enter **`StartSer RemoveInventory`** to start the Inventory Removal service.

The processing done by Inventory Removal service is as follows:

1. The Inventory Removal service reads each line of the inventoryremovallist.txt file and creates a delete str file for each inventoried server that is listed in the inventoryremovallist.txt file.

   The delete str file is saved in the scandir directory if the Selector is running, else it will be placed in the dbdir or entmergedir directories depending on the Inventory server role.

2. The Selector validates the delete str file and copies it into the dbdir and entmergedir directories.

3. The Storer reads the delete str file from dbdir and deletes the inventoried server from the attached Inventory database.
4. If the inventory deployment rolls up inventory information, the delete str is also rolled up to the next level Inventory server.

   The inventoried server is deleted from the Inventory database at all Inventory servers deployed at the enterprise level.

17 Viewing Inventory Information

This section provides information on the following topics:

Viewing the Inventory Information Using ConsoleOne

The following sections will explain the various types of information you can view using ConsoleOne:

- You can list hardware and software components found on the inventoried server and any custom information you have specified for the inventoried server.

  The Inventory Summary window displays the inventory items for an inventoried server. This window displays the information from the last inventory scan for the inventoried server. For more information, see "Viewing the Inventory Summary of an Inventoried Server" on page 620.

- You can list inventoried servers with the inventory information from the Inventory database satisfying the criteria you specify in the Inventory Query window. You form a query by specifying the component and its attribute for servers within the selected database sites.

  For more information about querying the Inventory database, see "Viewing Inventory Information of Inventoried Servers by Querying the Database" on page 632.

- You can use a list of reports that generate the inventory information from the Inventory database specific to your needs.

  For more information, see "Running Inventory Reports" on page 635.

Configuring the Inventory Database

If you want to view the inventory information stored in the database from ConsoleOne, you must configure the database. The inventory information from the Inventory database that you configure will be used for generating inventory reports, viewing inventory information, and for querying the inventory information from the database.

To configure the Inventory database:

1. In ConsoleOne, select a container.
2. Invoke Configure DB.
   - To invoke Configure DB from a database object, right-click the database object, click ZENworks Inventory, then click Configure DB. This configures the database object.

- To invoke the Configure DB dialog box from the ConsoleOne Tools menu, click Tools, click ZENworks Inventory, then click Configure DB.

3 Click Browse to browse for and select the ZENworks Database object.

You can also select an existing ZENworks Database object from the list of Database objects.

This Database object contains the database settings such as the protocol, port in use by the database, and others.

4 To apply this database configuration to all the sessions, select the Apply Configuration Across Sessions check box.

5 Click OK.

The database you configured is used for data retrieval unless you change it again using this same procedure.

Viewing the Inventory Summary of an Inventoried Server

The Inventory Summary window displays the information from the last inventory scan for the inventoried server.

To view the inventory information of an inventoried server, do the following in ConsoleOne

1 Configure the Inventory database.

For more information, see “Configuring the Inventory Database” on page 619.

2 Right-click any of the following objects: Subscriber, Distributor, or External Subscriber, click Actions, then click Inventory.

or

In the Query Results window, double-click an inventoried server.

ZENworks 6.5 Server Management provides the following inventory information collected from the inventoried servers:

Scan Data Group	Scan Data Item	Description
Inventory Information	Inventory Server	Name of the Inventory server to which the scans are sent
	Last Scan Date	List of all inventoried servers that were scanned on or before the specified date and time
	Scan Mode	Mode used by the Inventory scanner to scan the inventoried server
	Version	Version number of the Inventory scanner
	General Dictionary Version	Version number of the General Dictionary
	Private Dictionary Version	Version number of the Private Dictionary
Hardware/Software Inventory > General > System Information	Asset Tag	Asset tag number that the ROM-based setup program creates
	Computer Model	Identifying information of the computer, for example, Compaq, Dell, and others.

Scan Data Group	Scan Data Item	Description
	Computer Type	Type of computer, such as IBM PC, and others
	Machine Name	DNS name of the inventoried server
	Management Technology	Technology available on the inventoried server such as DMI, WMI, and others
	Model Number	Model number of the computer
	Serial Number	Serial number of the computer system assigned by manufacturer
	Tag	Unique identifier of system information
Hardware/Software Inventory > General > System Identification	Primary Owner Name	The name of the primary user or owner of this system
	Primary Owner Contact	The phone number of the primary user of this system
	Name	Name of the inventoried server as represented in eDirectory, such as the fully qualified DN of the inventoried server
Hardware/Software Inventory > General > Login Details > eDirectory Login Details	Current login user	User logged in to the Primary eDirectory tree when the inventoried server was scanned
	Last login user	User most recently logged into the Primary eDirectory tree through Novell Client when the inventoried server was scanned
Hardware/Software Inventory > General > Login Details > Windows Domain	Name	Domain name of the inventoried server
Hardware/Software Inventory > Software > Application Vendors > *Vendor_name* > *software_group_name* > *software*	Name	Vendor-defined name of the product represented as a vendor trademark or registered trademark
	Version	User-friendly version of a product. For example, the version for Windows 2000 is 2000 or Major.Minor Version of the Product
	Category	Product category to which the product belongs For example, Office is a part of the Productivity tools category and Solitaire is a game
	Description	Description of the product
	Help Link	Support web site URL for the product that is available in ARP and MSI

Scan Data Group	Scan Data Item	Description
	Package GUID	Vendor-defined GUID for a product that is available in MSI
	Product Identifier	A unique, 16-character identifier for an installed product. This identifier is available from MSI on Windows. The format is ABCD-1234-WXYZ-PQRS
	Internal Version	Internal version of a product The format is: *major version.minor version.build.sub build number* or *major version.minor version.build*
	Language	User-friendly name for the language of this copy of the product
	Uninstall String	The command to invoke for uninstalling this product instance. Currently, this is available in Add-Remove Programs (ARP) and MSI on Windows.
	Install Source	Identifies the file system path where the installation files were stored when installing this product instance. Currently, this is available in ARP and MSI on Windows.
	Last Execution Time	Date and time stamp when the product was last executed
	Frequency of Usage	Number of times the product is used
	Friendly Name	Display name of the software
	Installation Repository	Source of scan, which can be, Add-Remove Programs, MSI, Software Dictionary, or PRODUCTS.DAT
	Support Pack	Installed support pack number of the product
	Product Edition	Product edition defined by the vendor. For example, Professional
	Path	Directory path where the product is installed on the computer system
	AntiVirus Definition Date NOTE: This is applicable only for antivirus products.	The date of the virus definition file installed on the computer. Some anti-virus products combine date and version into a single string.
	AntiVirus Definition Version NOTE: This is applicable only for antivirus products.	The vendor-defined version of the virus definition file that has been installed on a computer.

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Software > Application Vendors > *Vendor_name* > *software_group_name* > *software* > Patches	Name	Vendor-defined name for the patch
Hardware/Software Inventory > Software > Application Vendors > *Vendor_name* > *software_group_name* > *software* > Representative File Information	File Name	Name of the file representing the software
	File Version	Version of the file representing the software
	File Size	Size of the file representing the software
	Last Modified	Last modified date of the file representing the software
	Internal Name	Internal name
	Product Version	The version of the product represented by this file
	Company	Vendor name
	Product Name	The product which this file represents
	Language	User-friendly name for the language of this copy of the file
	File Path	Location of the file on the inventoried server
	Software Dictionary ID	ID of the file as represented in the General software dictionary
Hardware/Software Inventory > Software > Disk Usage	File Extension Name	The file extension for which the disk usage is scanned for.
	Total Disk Usage	Total disk usage for all the files of the specified extension.
Hardware/Software Inventory > Software > Device Drivers > Pointing Device Drivers > *Pointing Device driver name*	Name	Name of the mouse driver
	Version	Version number of the mouse driver
Hardware/Software Inventory > Software > Device Drivers > Display Drivers	Install Date	Install date of the display driver
	Manufacturer	Name of the display driver manufacturer

Scan Data Group	Scan Data Item	Description
	Is Shadowed (True or False)	If True, the display driver is currently being shadowed
	Version	Version number of the display driver
Hardware/Software Inventory > Software > Device Drivers > Network Drivers	Description	Description of the network driver
	Name	Network driver name
	Version	Version number of the network driver
Hardware/Software Inventory > Software > Operating System	Code Page	Language code page of the operating system
	OS Type	Operating system of the inventoried server
	Install Date	Install date of the operating system
	Caption	Operating system name, for example, Windows 95/ Windows 2000
	Other Description	Additional description of the operating system if available
	Role	Type of the operating system such as server or workstation
	Total Virtual Memory Size	Total number of bytes in the virtual address space of the calling process
	Total Memory Size	Total memory of the operating system
	Version	Version number of the operating system
Hardware/Software Inventory > Hardware > Monitor	Device ID	Unique ID of a desktop monitor that is attached to a computer system For example, DesktopMonitor1
	Description	Description of the monitor.
	Nominal Size	A number representing the diagonal width of the monitor (the distance from one corner of the screen to the opposite corner of the screen) For example, 17” You can customize the scan of the nominal size of the monitor by configuring the HWRules ini file using the Server Inventory policy.
	Viewable Size	A number representing the diagonal width of the screen image excluding the black borders around the image’s edge For example, 15.8”

Scan Data Group	Scan Data Item	Description
	Manufacturer	Name of the monitor’s manufacturer For example, DELL* Computer Corp
	Serial Number	Manufacturer’s number used to identify a monitor For example, 23DDC24N9067
	Model	Product name of the monitor given by the manufacturer For example, DELL E771a
	Manufacture Date	Year in which the monitor was manufactured For example, 2003
	Model ID	Unique ID of a model of the monitor. It is a combination of the Manufacturer ID and Product ID For example, DELA001
Hardware/Software Inventory > Hardware > Chassis	Asset Tag	Asset tag number of the system chassis For example, S11127
	Number of Power Cords	Total number of power cords attached to a system chassis
	Chassis Type	Represents whether the system chassis is a laptop, desktop, notebook, docking station and so on
	Manufacturer	Name of the system chassis manufacturer For example, Compaq*
	Serial Number	Manufacturer’s number used to identify a system chassis For example, 53R661S
	Version	Version number of the system chassis
	Tag	Unique ID of the system chassis attached to a particular computer system For example, System Enclosure 0
Hardware/Software Inventory > Hardware > Pointing Device > *Pointing device name*	IRQ Number	Interrupt assigned to this device
	Name	Identifying information of the mouse
	Number of Buttons	Number of buttons on the mouse
Hardware/Software Inventory > Hardware > Keyboard	Delay	Delay before the repeat of a key

Scan Data Group	Scan Data Item	Description
	Description	Description of the keyboard, such as IBM Enhanced 101 or 102 keys
	Layout	Layout of the keyboard
	Number of Function Keys	Total number of function keys
	Subtype	Type of the keyboard
	Typematic Rate	Rate of processing the keys
Hardware/Software Inventory > Hardware > Display Adapter > *Display adapter name*	Chip Set	Chip set used by the controller to compare system capabilities
	Current Bits/Pixel	Number of adjacent color bits for each pixel
	Current Horizontal Resolution	Number of horizontal pixels shown by the display
	Current Vertical Resolution	Number of vertical pixels shown by the display
	DAC Type	Digital-to-Analog converter type
	Description	Description of the display adapter
	Maximum Memory Supported	Maximum memory that the display adapter supports for VIDEO RAM
	Maximum Refresh Rate	Maximum refresh rate of the monitor for redrawing the display, measured in Hertz
	Minimum Refresh Rate	Minimum refresh rate of the monitor for redrawing the display, measured in Hertz
	Number of Color Planes	Number of color planes supported by the video system
	Provider	Vendor name
	Video Architecture	The architecture of the video subsystem in this system, for example, CGA/VGA/SVGA/8514A
	Video Memory Type	The type of video memory for this adapter, for example, VRAM/SRAM/DRAM/EDO RAM
Hardware/Software Inventory > Hardware > BIOS	BIOS Identification Bytes	Byte in the BIOS that indicates the computer model
	Install Date	The manufacturing date of the BIOS
	Manufacturer	BIOS vendor name
	Caption	BIOS label
	Primary BIOS	True state indicates Primary BIOS

Scan Data Group	Scan Data Item	Description
	Serial Number	Serial number of the computer, assigned during manufacture
	Size	Size of the BIOS
	Version	Version or revision level of the BIOS
Hardware/Software Inventory > Hardware > Processor	Current Clock Speed (in MHz)	Current clock speed of the processor
	Device ID	Special hexadecimal string identifying the processor type
	Maximum Clock Speed (in MHz)	Maximum clock speed of the processor
	Other Family Description	Additional description about the Processor Family, such as Pentium Processor with MMX technology
	Processor Family	Identification of the processor family such as Pentium II, Pentium III, and others
	Processor Stepping	Single-byte code characteristic provided by microprocessor vendors to identify the processor model
	Role	Type of processor such as central processor, math coprocessor, and others
	Upgrade Method	The method by which this processor can be upgraded, if upgrades are supported
Hardware/Software Inventory > Hardware > Modem	Description	Additional information about the modem
	Name	Identifying information of the modem
	Device ID	Special hexadecimal string identifying the modem type
	Provider	Name of the vendor
Hardware/Software Inventory > Hardware > Battery	Chemistry	The battery chemistry, for example, lithium-ion or nickel metal hydride
	Design Capacity	The design capacity of the battery in mWatt-hours
	Design Voltage	The design voltage of the battery in mVolts
	Install Date	The battery manufacture date
	Manufacturer	The name of the company that manufactured the battery
	Name	Device name for this battery, for example, Duracell* DR-36
	Serial Number	The serial number for this battery

Scan Data Group	Scan Data Item	Description
	Smart Battery Version	The Smart Battery Data Specification version number supported by this battery
Hardware/Software Inventory > Hardware > Power Supply	Description	Expanded description of the input voltage capability for this power supply
	Total Output Power (in MilliWatts)	Attribute value that represents the total output power of the power supply
Hardware/Software Inventory > Hardware > Memory	Total Memory	Total memory of the inventoried server
Hardware/Software Inventory > Hardware > Disk Drives > Floppy	Capacity	Floppy drive capacity
	Description	Floppy drive description
	Drive Letter	Letter name of the drive
	Manufacturer	Vendor name
	Physical Cylinders	Floppy drive cylinders
	Physical Heads	Floppy drive R/W heads
	Sectors/Track	Floppy drive sectors per track
Hardware/Software Inventory > Hardware > Disk Drives > Physical Disk > Fixed Disk	Description	Description
	Manufacturer	Vendor name
	Physical Cylinders	Number of cylinders
	Physical Heads	Number of heads
	Sectors/Track	Fixed disk drive sectors per track
	Size	Size of the fixed disk
Hardware/Software Inventory > Hardware > Disk Drives > Physical Disk > Removable Disk	Description	Description
	Manufacturer	Vendor name
	Physical Cylinders	Number of cylinders
	Physical Heads	Number of heads
	Sectors/Track	Removable disk drive sectors per track
	Size	Size of the removable disk

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Hardware > Disk Drives > Logical Disk > *Logical disk name*	Drive Letter	Letter name of the drive
	File System Type	Type of file system, such as File Allocation Table (FAT)
	Free Size	Drive's actual size in MB
	Volume Label	Name of the hard disk volume
	Size	Drive's available space in MB
	Volume Serial Number	Hard disk volume serial number
Hardware/Software Inventory > Hardware > Disk Drives > CDROM	Name	Name of the CD drive attached to the inventoried server
	Description	Description of the CD
	Drive Letter	Mapped drive name of the CD
	Manufacturer	Vendor name
	Caption	Caption of the CD
Hardware/Software Inventory > Hardware > Ports > Serial Port	Address	Base input-output address for this serial port
	IRQ Number	IRQ number of the serial port
	Name	The logical name of the I/O device on this serial port, under this operating environment
Hardware/Software Inventory > Hardware > Ports > Parallel Port	Address	Base I/O address for this parallel port
	DMA Support (True or False)	If True, DMA is supported
	Name	The logical name of the input-output device on this parallel port, under this operating environment
	IRQ Number	IRQ number of the parallel port
Hardware/Software Inventory > Hardware > Bus	Bus Type	Bus type indicates PCI, ISA, and others
	Description	Bus description
	Name	Bus name
	Version	Version of the bus supported by the motherboard

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Hardware > Network Adapter	Adapter Type	Type of network adapter, such as FDDI or token ring
	Auto Sense	A Boolean value indicating whether the network adapter is capable of automatically determining the speed or other communication characteristics of the attached network media
	Card Manufacturer	Name of the card manufacturer
	Description	Adapter description
	Install Date	Install date of the network adapter
	Maximum Speed	Rate at which the information is transferred over the LAN
	Name	Network adapter name
	Permanent Address	Node address stored permanently in the adapter
	Provider	Name of the provider
Hardware/Software Inventory > Hardware > Sound Adapter	Description	Description of the multimedia component for the server
	Name	Label of the multimedia card
	Provider	Name of the provider
Hardware/Software Inventory > Network > DNS	DNS Name	The DNS name of the inventoried server
Hardware/Software Inventory > Network > Network (*instance_number)* > IP	IP Address	The unique address assigned to a computer on an IP Internet
	Subnet Mask	The subnet mask of the inventoried server paired with an IP address specifies to an IP router which octets or bits in the IP address are the network ID and which octets or bits are the node ID
Hardware/Software Inventory > Network > Network (*instance_number)* > IPX	IPX Address	The IPX™ address of the inventoried server
Hardware/Software Inventory > Network > Network (*instance_number)* > MAC	MAC Address	Unique node address permanently coded in the network adapter that identifies a specific computer on a network
Hardware/Software Inventory > Network > IP	IP Address	The unique address assigned to a computer on an IP Internet

Scan Data Group	Scan Data Item	Description
	Subnet Mask	The subnet mask of the inventoried server paired with an IP address specifies to an IP router which octets or bits in the IP address are the network ID and which octets or bits are the node ID
Hardware/Software Inventory > Network > IPX	IPX Address	The IPX address of the inventoried server
Hardware/Software Inventory > Network > MAC	MAC Address	Unique node address permanently coded in the network adapter that identifies a specific computer on a network
Hardware/Software Inventory > System > System IRQ	Availability	Availability of the specific IRQ channel
	IRQ Number	Number of the Interrupt Request Line (IRQ), from 0 to 15
	IRQ Trigger Type	IRQ Trigger type
	Shareable	If True, the system IRQ can be shared across devices
Hardware/Software Inventory > System > System Cache	Associativity	Defines the system cache associativity (direct-mapped, 2-way, 4-way)
	Cache Type	Defines the system cache type, for example, Instruction, Data, Unified
	Capacity	Size of the data store where the cache information is kept
	Error Methodology	Error correction scheme supported by this cache component, for example, Parity/Single Bit ECC/ MultiBit ECC
	Level	Indicates the cache level; internal cache that is built in to the microprocessors; external cache that is between the CPU and DRAM
	Line Size	Size in bytes of a single cache bucket or line
	Read Policy	Indicates whether the data cache is for read operations
	Replacement Policy	Algorithm that the cache uses to determine which cache lines or buckets should be reused
	Speed	Speed of this System Cache module in nanoseconds
	Write Policy	Indicates the two different ways (Write-Back and Write-Through Cache) that the cache can handle to write to the memory
Hardware/Software Inventory > System > System DMA	Availability	Indicates whether Virtual Direct Memory Access (DMA) is supported

Scan Data Group	Scan Data Item	Description
	Description	Name of the logical device that is currently using this DMA channel
	DMA Burst Mode	A data transmission mode in which data is sent faster than normal
	DMA Channel Number	Number of the Direct Memory Access (DMA) channel that a computer uses for transferring data to and from devices quicker than from computers without a DMA channel
Hardware/Software Inventory > System > System Slot	Description	Card currently occupying this slot
	Maximum Data Width	Maximum bus width of cards accepted in the slot
	Thermal Rating	Maximum thermal dissipation of the slot in milliwatts
Hardware/Software Inventory > System > Motherboard	Manufacturer	Name of the motherboard manufacturer
	Number of Slots	The number of expansion slots in the motherboard for adding more memory, graphic capabilities, and support for special devices
	Version	Version of the motherboard
	Description	General description of the motherboard

NOTE: For an enumerated attribute, the value will be displayed in the format *enumerated_value [enumerated_ID]*. For example, Processor.Processor Family = Pentium (R) III [17].

The Status bar displays the following information:

- **Tree Name:** Displays the eDirectory tree name where the inventoried server or inventoried server resides.
- **Recent Information:** Set to Yes if the Inventory database has been updated with the latest inventory information of the selected inventoried server.

Viewing Inventory Information of Inventoried Servers by Querying the Database

Using ConsoleOne, you can query the Inventory database to display the hardware and software components of inventoried servers that you want to view. The Inventory Query window displays the information satisfying the criteria you specify.

The Inventory database stores inventory information (general, hardware, software, network, and system information) for each inventoried server. Querying the Inventory database helps to create groups of similar devices and to focus your reports on specific types of machines. For example, you can query the database to find machines that have an i486D processor and a VGA card.

To query the Inventory database for inventory information:

1. In ConsoleOne, click a container.

2 Invoke Query.

- To invoke the Inventory query from a database object, right-click the database object, click ZENworks Inventory, then click Inventory Query.
- To invoke the Inventory query from the ConsoleOne Tools menu, you must first configure the database and then click Tools, click ZENworks Inventory, then click Inventory Query. For more information on how to configure the Inventory database, see "Configuring the Inventory Database" on page 619.

3 Specify the criteria for query:

Query the Inventory database for: By default, the Servers option will be enabled. The query locates all inventoried servers satisfying the query expression. If ZENworks 6.5 Server Management and Desktop Management are installed in the same environment; the Workstations, the Servers and the Both options will be available. When you select Servers, the query locates all inventoried servers satisfying the query expression. Choose Both to include all workstations and inventoried servers satisfying the query expression.

Find Type: Select Quick or Advanced. Click Quick to specify a simple query. When you choose a Quick query, you specify one attribute, relational operators, and the value of the attribute. Choose Advanced query to specify many attributes. You can combine multiple query groups so each group defines a set of query criteria. For example, use the Advanced query to run a query to discover all devices in the database with 486 processors and use query connectors, and add another query to discover which of these inventoried servers have a VGA color video adapter.

Display Machine(s) Not Satisfying the Query: Select the check box to retrieve machines that do not satisfy the query.

Select Attribute: Select the component or component attributes. Attributes that you can specify to query on the inventoried servers are grouped into the following categories: General, Software, Hardware, Network, and System.

The custom attribute will be prefixed by an asterisk (*).

For example, to find the machines that do not have pointing device installed, select Pointing Device as the component. To specify the version of BIOS as a component in the query, select BIOS as the component and VERSION as the component attribute.

Operator or Relational Operator: Select to determine the relationship between the components and the value. The relational operators are grouped on the basis on the data type of the attribute selected in the Select Attribute window as shown in the following table:

Data Type of the Attribute	Relational Operators
String	Equal To (=), Not Equal To (!=), Matches ([ ]), Does Not Match (![ ]) and Is NULL (null)
Numeric	Equal (=), Not Equal (!=), Less Than (<), Less Than or Equal To (< =), Greater Than (>), Greater Than or Equal To (>=), and Is NULL (null)
Date	After (>), On or After (>=), Before (<), On or Before (< =), and Is NULL (null)
Enum	Equal To (=), Not Equal To (!=), and Is NULL (null)
Custom	Includes all the relational operators that are grouped under the String, Numeric, and Date data types

NOTE: If the query does not display the result when the data type of the attribute is Custom and the relational operator is Numeric or Date, use the Equal To operator to find the values for the custom attributes that are stored in the Inventory database.

If you select only the component in the Select Attribute window, the Relational Operator will be set to NULL by default and other relational operators will not be available.

Value: Description values are the possible values of an inventory component. For example, 6.0 is a possible value for the DOS-Version attribute. Description values are not case sensitive.

NOTE: For an enumerated attribute, the value will be displayed in the format, *enumerated_value [enumerated_ID]*. For example, Processor.Processor Family = Pentium (R) III [17].

If you choose Matches ([ ]) or Does Not Match (![ ]) as the relational operator, you can use wildcards to substitute characters in the Value field. The following table lists the wildcards that can be used according the SQL documentation:

Example	Specifies to Include
?	Any one character
_ (underscore)	Any one character
%	Any string of zero or more characters
[]	Any one character in the specified range or set
[^]	Any one character not in the specified range or set

NOTE: To define a query using special characters such as ? or [, specify the query in the following formats: [?] or [[ ].

The list of description values displayed for an Inventory component is taken from the Inventory database corresponding to the component.

Logical Operator: This option is available only for the Advanced query. Logical Operator forms query groups that will be combined with the previous query group by using the relational operator specified between the query groups.

Save: This option is available only for the Advanced query. It saves the query expression as a file in the location that you specify. The query file does not have a default extension; however, we recommend the .qry extension for easy reference.

Load: This option is available only for the Advanced query. It loads the query file that you specify. You must provide the full filename with its extension.

4 Click Find.

This will query based on the query criteria you specify and display the inventoried servers that match the query in the Query Results window.

In the Query Results window, double-click the inventoried server or click File, then click Advanced Query to view the inventory information of the inventoried server.

Usage of Relational Operators

- **Match:** Use the Match operator to find the inventoried servers that satisfy the query condition.

  For example, use the Match operator to find all the inventoried servers with IP address 164.99.151.%.

- **NULL:** Use the NULL operator to query for those inventoried servers whose particular attribute is not scanned but the component has been scanned and some attributes are populated.

  For example, to find a list of inventoried servers for which BIOS.Manufacturer is not scanned, form a BIOS.Manufacturer is NULL query. This query will display the inventoried servers for which the BIOS has been scanned.

- **NOT SATISFYING:** Use the NOT SATISFYING query (or the NOT SATISFYING filter condition) to find filter conditions for the inventoried servers that negate the given query.

  For example, two servers S1 and S2 contain serial ports COM1 and COM2. The query (SerialPort='COM1') will return S1 and the query (SerialPort!='COM1') will also return the S1 because S1 contains the serial port COM2. To query the inventoried servers that do not contain the serial port COM1 you must use <NOT SATISFYING>(SerialPort='COM1'). To use the NOT SATISFYING option, click the Display Machines Not Satisfying the Query check box in the query window.

Running Inventory Reports

You can run reports to gather inventory information from the Inventory database. The Inventory reports are designed using Crystal Reports.

You can select from a predefined set of report forms to generate a report. The inventory report is displayed in the Crystal Viewer window.

You can print or export the report as desired. Remember that any reports you generate will be empty if you have not configured ZENworks 6.5 Server Management to start populating the Inventory database with the information you want.

This section covers information on the following sections:

- "Prerequisites for Generating Inventory Reports" on page 635
- "Types of Inventory Reports" on page 636
- "Generating Inventory Reports" on page 642
- "Printing an Inventory Report" on page 643
- "Exporting an Inventory Report to a File" on page 644
- "Running Inventory Reports" on page 635
- "Understanding User-Defined Reports" on page 644

Prerequisites for Generating Inventory Reports

Before running the inventory reports, ensure that you have installed the appropriate ODBC client. For more information, see *Novell ZENworks 6.5 Server Management Installation Guide*.

Types of Inventory Reports

You can generate the types of reports described below, assuming you have already configured ZENworks 6.5 Server Management to start populating the inventory database with the information you want.

The following table gives the Simple Inventory lists that provide information on individual aspects of Server Inventory, such as the operating system and the selection criteria. The table also lists the Comprehensive Inventory Reports that combine several aspects of Server Inventory into each report, such as memory, hard disk, and processor.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
Hardware Inventory	Asset Management Report	Scope, Machine Name, IP Address, and DNS Name You can also select to display the following options in the report: Memory, Processor, Display Adapter, Keyboard, Pointing Device, Fixed and Removable Disk, Floppy, CDROM, Network Adapter, and Monitor	Memory, processor, display details, keyboard, pointing device, fixed and removable disk, floppy, CD drive, network adapter, and monitor details for inventoried servers.
	BIOS Listing	Scope, Machine Name, IP Address, DNS Name, BIOS Install Date, and Manufacturer	List of all the inventoried servers with BIOS manufacturer, BIOS release date, and the total number of such machines.
	Battery Listing	Scope, Machine Name, IP Address, DNS Name, and Name	List of all inventoried servers that match the specified battery name.
	Bus Listing	Scope, Machine Name, IP Address, DNS Name, and Bus Type	List of all inventoried servers with the selected bus type.
	CDROM Listing	Scope, Machine Name, IP Address, DNS Name, Caption, Description, and Manufacturer	List of all inventoried servers that match the specified CD caption, description, and manufacturer's name.
	Display Adapter Listing	Scope, Machine Name, IP Address, DNS Name, Video Architecture, and Description	List of all inventoried servers that match the specified display adapter's video architecture and description.
	Floppy Listing	Scope, Machine Name, IP Address, DNS Name, Manufacturer, and Description	List of all inventoried servers that match the specified floppy description and manufacturer's name.
	Hardware Summary Report	Scope, Machine Name, IP Address, DNS Name, Operating System Type, Operating System Version, Processor Family, Curr. Clock Speed (Lower Bound in MHz), Curr. Clock Speed (Upper Bound in MHz), Total Memory (Lower Bound in MB), Total Memory (Upper Bound in MB), Hard Disk Size (Lower Bound in GB), and Hard Disk Size (Upper Bound in GB)	Operating system name, operating system version, processor family, processor current clock speed, memory, and hard disk size for each inventoried server.
	Keyboard Listing	Scope, Machine Name, IP Address, DNS Name, Description, and Layout	List of all inventoried servers that match the specified keyboard description and layout.
	Modem Listing	Scope, Machine Name, IP Address, DNS Name, and Name	List of all inventoried servers that match the specified modem name.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Monitor Listing	Scope, Machine Name, IP Address, DNS Name, Manufacturer, Manufacture Date, Nominal Size (Lower Bound in inches), and Nominal Size (Upper Bound in inches)	List of all inventoried servers that match the specified monitor manufacturer's name, manufacture date, and the specified range of monitor's nominal size.
	Network Adapter Listing	Scope, Machine Name, IP Address, DNS Name, and Name	List of all inventoried servers that match the specified network adapter's name.
	Physical Disk Listing	Show Chart, Scope, Machine Name, IP Address, DNS Name, Removable, Manufacturer, Description, Total Size (Lower Bound in GB), and Total Size (Upper Bound in GB)	List of all inventoried servers that match the specified physical disk manufacturer's name, description, the specified range of total size and disks that are fixed, removable, or both. You can also check the Show Chart box to display the Physical Disk Listing report in a pie chart.
	Pointing Device Listing	Scope, Machine Name, IP Address, DNS Name, Pointing Device Type, and Pointing Device Name	List of all inventoried servers that match the specified pointing device type and name.
	Power Supply Listing	Scope, Machine Name, IP Address, DNS Name, and Description	List of all inventoried servers that match the specified power supply description.
	Processor Listing	Show Chart, Scope, Machine Name, IP Address, DNS Name, Processor Family, Maximum Speed (Lower Bound in MHz), Maximum Speed (Upper Bound in MHz), Current Speed (Lower Bound in MHz), and Current Speed (Upper Bound in MHz)	List of all the inventoried servers with a processor family (such as Pentium Pro), processor maximum clock speed, and the processor current clock speed of the machines. You can also check the Show Chart box to display the Processor Listing report in a pie chart.
	Sound Adapter Listing	Scope, Machine Name, IP Address, DNS Name, and Name	List of all inventoried servers that match the specified sound adapter name.
	Storage Devices Inventory Report	Scope, Machine Name, IP Address, and DNS Name You can also select to display the following options in the report: Fixed and Removable Disk, Logical Disk, Floppy, and CDROM.	Fixed disk, removable disk, logical disk, floppy, and CD drive details for each inventoried server.
	System Chassis Listing	Scope, Machine Name, IP Address, DNS Name, Chassis Type, and Manufacturer	List of all inventoried servers that match the specified system chassis type and manufacturer's name.
System Configuration Inventory	Inventory Scan Listing	Show Chart, Scope, Machine Name, IP Address, DNS Name, Last Scan Date (On or Before), Inventory Server Name, and Recent Information	Date and time of the last inventory scan, Inventory server name, and recent information on each inventoried server. You can also check the Show Chart box to display the System Configuration Inventory report in a pie chart.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Memory Listing	Show Chart, Scope, Machine Name, IP Address, DNS Name, Total Memory (Lower Bound in MB), and Total Memory (Upper Bound in MB)	List of all the inventoried servers within a range of memory size (such as 200-400 MB) and the total number of such machines. You can also check the Show Chart box to display the Memory Listing report in a pie chart.
	Operating System Listing	Show Chart, Scope, Machine Name, IP Address, DNS Name, Operating System Type, and Operating System Version	List of all inventoried servers that match the specified operating system type and version. You can also check the Show Chart box to display the Operating System Listing in a pie chart.
	Networking Information Report	Scope, Machine Name, IP Address, and DNS Name	Network adapter type, DNS, IP address, MAC address, IPX address, and Windows Domain name for each inventoried server.
	System Information Listing	Scope, Machine Name, IP Address, DNS Name, and Computer Manufacturer	List of all inventories servers that match the specified computer manufacturer's name.
	System Internal Hardware Inventory Report	Scope, Machine Name, IP Address, and DNS Name You can also select to display the following options in the report: System IRQ, System Cache, System DMA, System Slot, and Motherboard.	IRQ, cache, DMA, slot, and motherboard for each inventoried server.
Software Inventory	Add-Remove Programs by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Software Name	List of all software that are listed in the "Add-Remove Programs" list for each inventoried server.
	Anti-Virus Signature Files by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Software Name, Min Virus Definition Date and Max Virus Definition Date	List of all antivirus signature files grouped by antivirus product installed on each inventoried server.
	Anti-Virus Signature Machine Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Software Name, Min Virus Definition Date and Max Virus Definition Date	List showing the count of inventoried servers that have any antivirus product installed.
	Disk Usage by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and File Extension	List of all inventoried servers and the disk usage that match the specified file extension.
	Exception List by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, File Name, Vendor Name, and Product Name	List of all inventoried servers and the file information that match the specified filename, vendor name, and product name.
	Installed NetWare Software by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Software Name	List of all inventoried NetWare machines and the products.dat details that match the given software name.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Internet Explorer Installation Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Internet Explorer Version, and Service Pack	List showing the count of inventoried servers with Internet Explorer installed.
	Internet Explorer Patches by Machine	Scope, Machine Name, IP Address, DNS Name, Internet Explorer Version, and Service Pack	List of all installed patches for the Internet Explorer version that matches the specified value and patch name.
	Internet Explorer by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Internet Explorer Version, and Service Pack	List of all Internet Explorer installations that match the specified version.
	MSI Products by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Software Name, and Vendor Name	List of all products installed on each inventoried server and that are listed in the MSI (Microsoft Installer) database.
	Microsoft Office Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Microsoft Office Version, and Service Pack	List of all products that match the specified product name and vendor name, and have been installed from the specified source.
	Microsoft Office Installation Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Microsoft Office Version, and Service Pack	List showing the count of inventoried servers with Microsoft Office installed.
	Microsoft Office by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Microsoft Office Version, and Service Pack	List of all Microsoft Office installations that match the specified version.
	Novell Client Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Novell Client Version, and Service Pack	List of all Novell Client components that match the specified version.
	Novell Client Installation Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Novell Client Version, and Service Pack	List showing the count of inventoried servers with Novell Client installed.
	Novell Client by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Novell Client Version, and Service Pack	List of all Novell Client installations that match the specified version.
	Outlook Express Installation Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Outlook Express Version	List showing the count of inventoried servers with Outlook Express installed.
	Outlook Express by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Patch Name	List of all Outlook Express installations that match the specified version.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Software Dictionary Application Files by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Vendor Name, Software Name, and Software Version	List of all inventoried servers and their software dictionary application files that match the specified vendor, software, and software version.
	Software Dictionary Applications by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Vendor Name, Software Name, and Software Category	List of all inventoried servers and their software dictionary applications that match the specified vendor, software, and software version.
	Software Dictionary Versions Machine Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, General Dictionary Version, and Private Dictionary Version	List showing the count of all inventoried servers with specified General Dictionary and Private Dictionary versions.
	Software Dictionary Versions by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, General Dictionary Version, and Private Dictionary Version	List of all inventoried servers with specified General Dictionary and Private Dictionary versions.
	Software Installation Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Vendor Name, Software Name, and Software Version	List showing the count of inventoried servers with specified vendor name, software, and version.
	Software Installations	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Vendor Name, Software Name, and Software Version	List of all inventoried servers with specified vendor name, software, and version.
	Software by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Vendor Name, Software Name, and Software Version	List of all inventoried servers and software information that match the specified vendor name, software, and version.
	System Software Inventory Report	Scope, Machine Name, IP Address, and DNS Name. You can also select to display the following options in the report: Display Driver, Pointing Device Driver, Network Adapter Driver, and NetWare Client.	Drivers (such as pointing device drivers, network adapter drivers, and display drivers) and Novell NetWare® Client for each inventoried server.
	Windows Components by Machine	Scope, Machine Name, IP Address, DNS Name, Windows Version, and Service Pack	List of all Windows components that match the specified version.
	Windows Installation Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Windows Version, and Service Pack	List showing the count of inventoried servers that have Windows operating system installed.
	Windows Media Player Count	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Windows Media Player Version	List showing the count of inventoried servers with Windows Media Player installed.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Windows Media Player Patches by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Windows Media Player Version	List of all patches for Windows Media Player installations that match the specified version and patch name.
	Windows Media Player by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, and Windows Media Player Version	List of all Windows Media Player installations that match the specified version.
	Windows Operating System by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Windows Version, and Service Pack	List of all Windows operating systems that match the specified version and serial number.
	Windows Security Patches by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, Max Last Scan Time, Windows Version, and Service Pack	List of all patches for Windows operating systems that match the specified version and patch name.
	Novell ZENworks Desktop Management Installed Agent Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the agent components of ZENworks 6.5 Desktop Management installed on these machines.
	Novell ZENworks Desktop Management Installed Server Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the server components of ZENworks 6.5 Desktop Management installed on these machines.
	Novell ZENworks Handheld Management Installed Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the ZENworks 6.5 Handheld Management components installed on these machines.
	Novell ZENworks Installed Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the ZENworks 6.5 components installed on these machines.
	Novell ZENworks Installed Suites by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the ZENworks 6.5 suites installed on these machines.

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Novell ZENworks Server Management Installed Agent Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the agent components of ZENworks 6.5 Server Management installed on these machines.
	Novell ZENworks Server Management Installed Server Components by Machine	Scope, Machine Name, IP Address, DNS Name, Min Last Scan Time, and Max Last Scan Time	List of all machines that were successfully last scanned within the specified time range and the server components of ZENworks 6.5 Server Management installed on these machines.
Others	User Defined Reports For more information on how to create user-defined reports, see the "Understanding User-Defined Reports" on page 644.	Based on the options specified by the user in the consoleone*consoleone_version*\bin\userreports.ini file	Displays the user-defined report.

NOTE: The Show Chart selection criteria display a graphical representation of the Inventory report.

Generating Inventory Reports

To generate the inventory report:

1. Invoke the Inventory report by using any of the following methods:
   - To invoke the Inventory report from a database object, right-click the database object, then click ZENworks Reports.
   - To invoke the Inventory report from the ConsoleOne Tools menu, you must first configure the database, click Tools, then click ZENworks Reports. For more information on how to configure the database, see "Configuring the Inventory Database" on page 619.
2. Click the report you want to generate.

   The description for the report is displayed on the right side of the screen.

   See the table with listing of simple Inventory lists and listing of the comprehensive inventory reports.
3. Specify the selection criteria.

   The Scope selection criteria are enabled only if both ZENworks 6.5 Desktop Management and ZENworks 6.5 Server Management are installed on the same machine.

For example, to view all inventoried servers that have the Windows 2000 operating system, you would select Operating System Listing, specify the selection criteria scope as Both, and the operating system type as Windows 2000. The report will display the inventory information of all servers within the configured Inventory database.

Depending on the type of report you want, you can filter the information. For example, to view all inventoried servers with the Windows operating system, you select the Operating System Listing, and specify the selection criteria Scope as Both, the Operating System Type as Windows, and the Operating System Version as 2000.

Follow these guidelines as you work with the Reporting dialog box:

- The selection criteria in the Inventory report are case sensitive.

  For example, if you want to know the list of machines whose Distinguished Name is CN=MACHINE1.OU=ENG.O=NOVELL, specify OU=ENG.O=NOVELL as the selection criterion. All the machines whose DN contains OU=ENG.O=NOVELL are displayed in the Inventory report, but the machines whose DN contains ou=eng.o=novell are not displayed in the Inventory report.

- If the Reporting dialog box allows wildcards, you can use an asterisk (*) or question mark (?) with all selection criteria. The wildcard characters can be used for character data only.

  The following table lists examples of wildcards usage.

Example	Specifies to Include
*	All items
164.99.*	All items starting with 164.99.
164.9?.215.23	All items starting with 164.9, followed by any character, and ending with ".215.23"
164.96.215.23	The single named item, in this case the inventoried server with the specified IP address

4 Click Run Selected Report.

A status box appears displaying the progress of the report generation. When the report is generated, it appears in the viewer. Use the buttons on the toolbar to page through, print, or export the report.

NOTE: ZENworks Inventory report supports only the following double-byte character languages: German, English, Spanish, French, Portugese, and Japanese. Other double-byte characters might not be displayed properly in the Inventory reports.

Printing an Inventory Report

To print a report:

1 Generate and view the report.

2 To change the default settings of the Printer, click the Printer Setup icon and modify the settings.

3 Click the Printer icon.

Exporting an Inventory Report to a File

To export an inventory report to a file:

1. Generate and view the report.
2. On the toolbar, click the Export Report icon  .
3. In the Export dialog box, specify the location and file format.

   If you choose to export the Inventory report to a text file, in the Export to Text dialog box, select the User defined option and set the value to 16 because the data exported will be truncated if the value is less than 16.

   If you want to export the Inventory report to an HTML file, you can select HTML 3.2 or HTML 4.0 (DHTML) file format. We recommend that you export to HTML 4.0 (DHTML) because the data exported to HTML 3.2 will not be formatted properly.

   If you want to export the Inventory report to a comma-separated value (.csv) file, do the following:

   3a. Export the report to Microsoft* Excel.

   NOTE: If you choose to export to .csv, the report will not be properly exported.

   3b. Open the .xls file.

   3c. Click File, then click Save As.

   3d. In the Save as type field, choose CSV (Comma delimited) (*.csv).

   3e. Click Save.
4. Click OK.
5. Browse for and select the directory where you want to save the exported file.
6. Click OK.

Understanding User-Defined Reports

Using the Crystal Report Designer you can generate reports with the information present in the Inventory database.

Before generating the reports, you must ensure that the report file (.rpt) is created using Crystal Report Designer 8.0/8.5. For more information on how to create a .rpt file, see the Crystal Report documentation.

IMPORTANT: Except for the Software Inventory reports, you can use any Inventory report as a template to create a report.

To generate the User-defined Inventory report:

1. On the machine where you are designing the report, set the ODBC DSN name to ZenInventory.

   To set the ODBC name:

   1a. Click Start, click Settings, then click Control Panel.

   1b. Double-click ODBC Data Sources (32 Bit), then click Add

   1c. Select the ODBC driver for the database you want to connect to.

   1d. Click Finish.

1e Specify the Data Source name as ZenInventory and specify the details.

NOTE: If you want to specify a data source name other than ZenInventory, you must configure the ODBC name on the each of the machines where you invoke user-defined reports through ConsoleOne.

2 After you have designed the report, place the report in the \consoleone\version\reporting\canned\novellreporting\zeninventory\en directory.

3 Set the values in the userreports.ini file in the \consoleone\version\bin directory. The userreports.ini file must contain the following values:

```
#[ReportName] <actual name of the report file without the .rpt extension>

#DisplayName=User Defined Report's display name

#Param1=Constant,Display name,<if combo then {val-1|val-2|val-3}>

#<where Param1 is the internal name of the parameter as stored in the .rpt
file>

#<Constants are 1, 2 and 3 for Combo selection, text field and numeric
field respectively>
```

For example, you can set the value as given below:

```
[ListSystemInformation]DisplayName=System Information

Role=1,Role,{2|3|5}

IPAddress=2,IP Address

DNName=2,Distinguished Name

DNTree=2,Distinguished Tree

DNSName=2,DNS Name

[ListMemory]

DisplayName=Memory

Role=1,Role,{2|3|5}

IPAddress=2,IP Address

DNName=2,Distinguished Name

DNTree=2,Distinguished Tree

DNSName=2,DNS Name

MemoryLowerLimit=3,Memory Lower Bound
```

4 After you set the values in the USERREPORTS.INI file, the User Defined Report is displayed in the Inventory Reports tree. You can specify multiple reports in the USERREPORTS.INI files.

NOTE: If the USERREPORTS.INI file is empty, the user cannot view the User Defined Reports in the Inventory Reports tree.

5 Click Run Selected Report.

Exporting the Inventory Information

You can customize the inventory information you want to export from the ZENworks 6.5 Server Management Inventory database in to a comma-separated value (.csv) or an XML file.

You select the inventory components that should be exported, such as the Operating System Name and Version. You can further filter the inventoried servers whose attributes will be exported. For example, you can export only those inventoried servers with a particular processor speed. The Data Export tool will export all inventoried servers satisfying these query conditions into a .csv or .xml file.

If you want to reuse the same data export settings for export, you can save the data export configurations.

The following sections will help you use the Data Export tool:

- "Procedure to Export the Inventory Information" on page 646
- "Loading an Existing Configuration File" on page 648
- "Running the Data Export Program from the Inventory Server" on page 650

Procedure to Export the Inventory Information

1. In ConsoleOne, select a container.
2. Invoke the Data Export tool.
   - To invoke the Data Export tool from a database object, right-click the database object, click ZENworks Inventory, then click Data Export.
   - To invoke the Data Export tool from the ConsoleOne Tools menu, you must first configure the Inventory database and then click Tools, click ZENworks Inventory, then click Data Export. For more information on how to configure the Inventory database, see "Configuring the Inventory Database" on page 619.
3. Select Create a New Database Query.

   This option lets you add a new query that defines the inventory components such as hardware, software, network, and others that you want to export. You can also specify the criteria to limit the inventoried servers and the database sites to be included in the query. Based on the inventory components and criteria you specify, the inventory information from the database is exported to a .csv or .xml file.

   NOTE: If you want to load existing configuration settings for data export, select Open a Saved Database Query. This options lets you modify the settings for data export and then export the data to a .csv or .xml file. For more information, see "Loading an Existing Configuration File" on page 648.
4. Click Next.
5. Specify the filter conditions for the inventoried servers.

   5a Click Edit Query. For more information on how to define a query, see "Viewing Inventory Information of Inventoried Servers by Querying the Database" on page 632.

   5b (Optional) The Enable Filter option is available for selection only if you define the query using the software classes and its attributes of a supported category. Following are supported categories:

   Category 1: Software Group, Software Group File Information, Software Group Patch Information, Software, File Information, and Patch Information

Category 2: Exclude File Information

Category 3: Disk Usage

The Enable Filter option will not be available for selection if the query contains attributes belonging to different categories. For example, a query containing software.name=word, softwaregroup.name=office, and diskusage.name=exe.

If you want the results that will be stored in .csv or .xml file to be filtered on the basis of the above query, select the Enable Filter check box.

5c Set the scope for exporting the information from the Inventory database.

If the ConsoleOne snap-ins and the Data Export tool have been installed for both ZENworks 6.5 Server Management and ZENworks 6.5 Desktop Management, the Data Export tool allows you to change the scope of exporting the inventory information.

By default, the Servers option will be enabled. The query locates all inventoried servers satisfying the query expression. If ZENworks 6.5 Server Management and ZENworks 6.5 Desktop Management are installed in the same environment, the Workstations, the Servers and the Both options will be available. When you select Servers, the query locates all inventoried servers satisfying the query expression. Choose Both to include all inventoried workstations and inventoried servers satisfying the query expression.

5d Review the query that you define.

5e Click Next.

6 Select the database fields from the list of database fields, then click Add.

If you select a group component, all subcomponents of the group are added. For example, if you select the Software component group, the subcomponents of Software such as vendor name, product name, and version are added.

7 Click Next.

8 View the data export settings.

8a Click Save Configuration to save the configurations settings to an .exp file. Specify the filename for the .exp file and then click Save.

The configuration file (.exp) contains the settings such as the inventory components you selected, and also the query formed for filtering the inventoried server data export. You create an .exp file so that you can reload the configuration settings and generate the .csv or .xml files any time you need to.

8b Click Next.

9 Select the machine from where you intend to perform the query.

9a **Perform the Query from This Computer:** Select Perform the Query from This Computer to run the data export processing from the computer. This option will access the Inventory database on the specified database server and export the data in to a .csv or .xml file.

Perform the Query on a Remote Server: Select Perform the Query on a Remote Server to run the data export program from any server that has Server Inventory components installed.

Running the Data Export program from a server is recommended if you are exporting information from a large database with more than 10,000 inventoried servers or if you have specified complex queries with more than 20 database fields selected for exporting.

9b If you want to apply default encoding of the machine to the .csv or .xml file, select Default Encoding. The Default Encoding check box is selected by default. To apply Unicode encoding to the .csv or .xml file, select Unicode Encoding.

NOTE: If you create an .exp file to perform the data export from the local machine but use the same .exp to perform data export from a remote server and you want Unicode encoding, you must manually edit the .exp file and set the value of DEExportEncode to UNICODE.

9c Click Next.

10 Select an export option.

10a Select one the following options:

Export to CSV: Saves the inventory information in a .csv file.

Export to XML: Saves the inventory information in a .xml file.

10b Specify the path and the filename where you want to create the .csv or.xml file.

10c Click Finish.

If the configuration settings are not saved, you will be prompted to save the changes

This generates the .csv or .xml file in the specified directory.

Open the .csv file in Microsoft Excel or any other CSV-supported viewer to view the exported data.

Open the .xml file in a XML viewer such as XML Spy. For more information, see "An Overview of XML and the Contents of an XML File" on page 650.

Loading an Existing Configuration File

You can load an existing configuration file (.exp). An .exp file contains the settings such as the inventory components you selected, and also the query formed for filtering the inventoried server data export.

After you load the .exp file, you can modify the settings for data export and then export the data to a .csv or .xml file.

To load existing configuration settings for data export:

1 Ensure that you have generated the data configuration files.

Complete the procedure outlined in "Procedure to Export the Inventory Information" on page 646. This procedure generates the .csv or .xml file and the data configuration files.

2 In ConsoleOne, select a container and invoke the Data Export tool using any of the following methods:

- To invoke the Data Export tool from a database object, right-click the database object, click ZENworks Inventory, then click Data Export.
- To invoke the Data Export tool from the ConsoleOne Tools menu, you must first configure the Inventory database and then click Tools, click ZENworks Inventory, then click Data Export. For more information on how to configure the Inventory database, see "Configuring the Inventory Database" on page 619.

3 Select Open a Saved Database Query, then click Next.

The default directory for .exp files is consoleone*consoleone_version*\reporting\export. Click Browse to open an existing .exp file.

If the .exp and .cfg files are invalid or are an older version, the data export will not proceed. The data export displays the number of servers and servers that satisfy the query and filter conditions for export.

4 Select a saved database query from the list of saved queries.

4a Select a saved database query from the list of saved queries. The list box displays the .exp files that are saved in consoleone*consoleone_version*\reporting\export.

or

Click Browse to open an existing .exp file in any other location.

4b (Optional) If the .exp and .cfg files are invalid or are an older version, the data export will not proceed. The data export displays the number of servers and servers that satisfy the query and filter conditions for export.

If you want to modify the existing query, click Edit and modify the query and select the new database fields. For more information on how to define a query, see "Viewing Inventory Information of Inventoried Servers by Querying the Database" on page 632.

4c Click Next.

5 View the data export settings.

5a Click Save Configuration to save the configurations settings to an .exp file. Specify the filename for the .exp file and then click Save.

The configuration file (.exp) contains the settings such as the inventory components you selected, and also the query formed for filtering the inventoried server data export. You create an .exp file so that you can reload the configuration settings and generate the .csv or .xml files any time you need to.

5b Click Next.

6 Select the machine from where you intend to perform the query.

6a **Perform the Query from This Computer:** Select Perform the Query from This Computer to run the data export processing from the computer. This option will access the Inventory database on the specified database server and export the data in to a .csv or .xml file.

Perform the Query on a Remote Server: Select Perform the Query on a Remote Server to run the data export program from any server that has Server Inventory components installed.

Running the Data Export program from a server is recommended if you are exporting information from a large database with more than 10,000 inventoried servers or if you have specified complex queries with more than 20 database fields selected for exporting.

6b If you want to apply default encoding of the machine to the .csv or .xml file, select Default Encoding. The Default Encoding check box is selected by default. To apply Unicode encoding to the .csv or .xml file, select Unicode Encoding.

6c Click Next.

7 Select an export option.

7a Select one the following options:

Export to CSV: Saves the inventory information in a .csv file.

Export to XML: Saves the inventory information in a .xml file.

7b Specify the path and the filename where you want to create the .csv or.xml file.

7c Click Finish.

Running the Data Export Program from the Inventory Server

Running the Data Export program from a server is recommended if you are exporting information from a large database with more than 10,000 inventoried servers or if you have specified complex queries with more than 20 database fields selected for exporting.

To run the data export program from the server:

1. Ensure that you have generated the data configuration files.

   Follow the Step 1 to Step 5 as outlined in "Procedure to Export the Inventory Information" on page 646 and ensure that you save the settings in the .exp file.

   When you save an .exp file, a corresponding data configuration file is created in the same directory with the same filename as the .exp file and with the .cfg file extension.

2. Click Perform the Query on a Remote Server to run the data export program from any server that has Server Inventory components installed, then click Finish.

3. Copy the .exp file and .cfg file to the server.

   These two files should exist in the same directory on the Inventory server.

   From the server console, run dbexport.ncf on NetWare servers or dbexport.bat on Windows servers. To do so, enter

   DBEXPORT *"configuration_filename.exp" "csv_filename.csv*"

   where *configuration_filename.exp* is an existing file that contains the data export settings. You must enter the *configuration_filename.exp* and the *csv_filename.csv* filenames within double quotes. The data exported from the database will be stored in *csv_filename.csv*.

   The corresponding .cfg file for the .exp file should be in the same folder as the .exp file. The .cfg file contains the list of the database attributes to be exported.

   If the .exp and .cfg files are invalid or are older versions, the data export will not proceed. The data export displays the number of inventoried servers that satisfy the query and filter conditions for export.

An Overview of XML and the Contents of an XML File

Server Inventory allows you to export the inventory information from the Inventory database into an Extensible Markup Language (.xml) file by using the Data Export tool.

XML is a markup language that provides a format for describing structured data. An XML document is a text-based format. The XML source is made up of XML elements. The XML tags are not predefined and you must define your own tags.

For more information about XML, see the World Wide Web Consortium (W3C) XML Activity and Information web site (http://www.w3.org/XML).

A sample .xml file is as follows:

```
<?xml version="1.0" encoding='UTF-8'?>

<!DOCTYPE InventoryInformation [<!ELEMENT Attribute (value)>

<!ATTLIST Attribute
```

```
     name CDATA #REQUIRED
     type (custom | regular) #REQUIRED
     units CDATA #IMPLIED
>
<!ELEMENT Class (Attribute*)>
<!ATTLIST Class
     name CDATA #REQUIRED
     instance CDATA #REQUIRED
>
<!ELEMENT InventoryInformation (Machine+)>
<!ELEMENT Machine (Class+)>
<!ATTLIST Machine
     name CDATA #REQUIRED
>
<!ELEMENT value (#PCDATA)>]
>
<InventoryInformation>
  <Machine name="blr-stl-zen1.blr.novell.com">
    <Class name="Processor" instance="1">
      <Attribute name="Current Clock Speed" type="regular" units="MHz">
        <value>2800</value>
      </Attribute>
      <Attribute name="Processor Family" type="regular">
        <value>"Intel(R) Xeon(TM)"</value>
      </Attribute>
    </Class>
    <Class name="IP" instance="1">
      <Attribute name="IP Address" type="regular">
        <value>164.99.163.9</value>
      </Attribute>
      <Attribute name="Subnet Mask" type="regular">
        <value>255.255.252.0</value>
      </Attribute>
    </Class>
  </Machine>
```

```
</InventoryInformation>
```

XML uses a Document Type Definition (DTD) to describe the data. DTD is embedded within the XML document.

A DTD lists the elements, attributes, and entities contained in a document and also, defines the relationship between the elements and attributes.

Following is the DTD embedded in the preceding sample xml file:

```
<?xml version="1.0" encoding='UTF-8'?>

<!DOCTYPE InventoryInformation [<!ELEMENT Attribute (value)>

<!ATTLIST Attribute

     name CDATA #REQUIRED

     type (custom | regular) #REQUIRED

     units CDATA #IMPLIED

>

<!ELEMENT Class (Attribute*)>

<!ATTLIST Class

     name CDATA #REQUIRED

     instance CDATA #REQUIRED

>

<!ELEMENT InventoryInformation (Machine+)>

<!ELEMENT Machine (Class+)>

<!ATTLIST Machine

     name CDATA #REQUIRED

>

<!ELEMENT value (#PCDATA)>]

>
```

The following table explains the elements used in the sample XML file:

Elements Used in the Sample XML File	Description
Class	Device name
Type	Custom or Regular attribute
Units	Unit information
Instance	Device instance count

Retrieving Inventory information from the Inventory Database Without Using the CIM Schema

ZENworks 6.5 Server Management SP1 provides easy-to-use Inventory database views that allow you to retrieve inventory information from the Inventory database without using the CIM schema.

The Inventory views are predefined device-specific views that are automatically created in the Inventory database after you install the Server Inventory component of ZENworks 6.5 Server Management SP1.

The nomenclature for the Inventory views is *database_schema_name.zen_devicename*. For example, mw_dba.zen_processor.

Inventory views that are associated with enums have localized views. For example, mw_dba.zen_processor_ja is the Japanese view for the Processor.

The following sections provide information about the various Inventory views and how to use them:

- "List of Inventory Views" on page 653
- "How to Use the Inventory Views" on page 675

List of Inventory Views

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_asset	Retrieves the Asset information	SystemName	DNS name of the inventoried server	Yes
		Manufacturer	Name of the manufacturer	
		Model	Model of the computer system	
		SerialNumber	Serial number of the computer system assigned by manufacturer	
		Tag	Unique identifier of system information	
		ManagementTechnology	Technology available on the inventoried server such as DMI, WMI, and others	
		AssetTag	Asset tag number that the ROM-based setup program creates	
		ModelNumber	Model number of the computer system	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_battery	Retrieves the Battery information	Name	Device name for the battery, for example, Duracell* DR-36	Yes
		Chemistry	The battery chemistry, for example, lithium-ion or nickel metal hydride	
		DesignCapacity	The design capacity of the battery in mWatt-hours	
		DesignVoltage	The design voltage of the battery in mVolts	
		SmartBatteryVersion	The Smart Battery Data Specification version number supported by this battery	
		InstallDate	The battery manufacture date	
		Manufacturer	The name of the company that manufactured the battery	
		SerialNumber	The serial number for the battery	
mw_dba.zen_bios	Retrieves the BIOS information	Caption	BIOS label	Yes
		InstallDate	The manufacturing date of the BIOS	
		SerialNumber	Serial number of the computer, assigned during manufacture	
		Version	Version or revision level of the BIOS	
		Manufacturer	BIOS vendor name	
		PrimaryBIOS	True state indicates Primary BIOS	
		BIOSIDBytes	Byte in the BIOS that indicates the computer model	
		Size	Size of the BIOS	
mw_dba.zen_bus	Retrieves the Bus information	BusType	Bus type indicates PCI, ISA, and others	Yes
		BusName	Bus name	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		BusDescription	Bus description	
		BusVersion	Version of the bus supported by the motherboard	
		DeviceID	The unique hexadecimal ID for the specific bus	
mw_dba.zen_cachememory	Retrieves the Cache memory information	ErrorMethodology	Error correction scheme supported by this cache component, for example, Parity/Single Bit ECC/ MultiBit ECC	Yes
		Level	Indicates the cache level; internal cache that is built in to the microprocessors; external cache that is between the CPU and DRAM	
		WritePolicy	Indicates the two different ways (Write-Back and Write-Through Cache) that the cache can handle to write to the memory	
		CacheType	Defines the system cache type, for example, Instruction, Data, Unified	
		LineSize	Size in bytes of a single cache bucket or line	
		ReplacementPolicy	Algorithm that the cache uses to determine which cache lines or buckets should be reused	
		ReadPolicy	Indicates whether the data cache is for read operations	
		Associativity	Defines the system cache associativity (directmapped, 2-way, 4-way)	
		Speed	Speed of this System Cache module in nanoseconds	
		Capacity	Size of the data store where the cache information is kept	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_cdrom	Retrieves the CDROM information	DeviceID	Drive letter allocated for the CD on the inventoried server	No
		Manufacturer	Vendor name of the CD	
		Description	Description of the CD	
		Caption	Caption of the CD	
mw_dba.zen_chassis	Retrieves the Chassis information	AssetTag	Asset tag number of the system chassis. For example, S11127	Yes
		NumberOfPowerCords	Total number of power cords attached to a system chassis	
		ChassisType	Represents whether the system chassis is a laptop, desktop, notebook, docking station and so on	
		Manufacturer	Name of the system chassis manufacturer. For example, Compaq	
		SerialNumber	Manufacturer's number used to identify a system chassis. For example, 53R661S	
		Tag	Unique ID of the system chassis attached to a particular inventoried server. For example, System Enclosure 0	
		Version	Version number of the system chassis	
mw_dba.zen_computerinformation	Retrieves the computer information	ComputerName	Name of the inventoried server as represented in eDirectory, such as the fully qualified DN of the inventoried server	No
		PrimaryOwner	The name of the primary user or owner of this system	
		PrimaryOwnerContact	The phone number of the primary user of this system	
mw_dba.zen_currentlogindetails	Retrieves the current login information	CurrentUser	User logged in to the Primary eDirectory tree when the inventoried server was scanned	No

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_disk	Retrieves the disk information	RemovableDisk	Removable disk	Yes
		Manufacturer	Vendor name of the disk	
		Description	Description of the disk	
		PhysicalCylinders	Number of cylinders	
		PhysicalHeads	Number of heads	
		SectorsPerTrack	Removable disk drive sectors per track	
		Capacity	Capacity of the removable disk	
mw_dba.zen_diskusage	Retrieves the disk usage information	FileExtension	The file extension for which the disk usage is scanned for.	No
		TotalDiskUsage	Total disk usage for all the files of the specified extension.	
mw_dba.zen_displayadapter	Retrieves the display adapter information	Description	Description of the display adapter	Yes
		VideoMemoryType	The type of video memory for this adapter, for example, VRAM/SRAM/ DRAM/EDO RAM	
		MaxMemorySupported	Maximum memory that the display adapter supports for VIDEO RAM	
		CurrentBitsPerPixel	Number of adjacent color bits for each pixel	
		CurrentHorizontalResolution	Number of horizontal pixels shown by the display	
		CurrentVerticalResolution	Number of vertical pixels shown by the display	
		MaxRefreshRate	Maximum refresh rate of the monitor for redrawing the display, measured in Hertz	
		MinRefreshRate	Minimum refresh rate of the monitor for redrawing the display, measured in Hertz	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		VideoArchitecture	The architecture of the video subsystem in this system, for example, CGA/ VGA/SVGA/8514A	
		NumberOfColorPlanes	Number of color planes supported by the video system	
		ChipSet	Chip set used by the controller to compare system capabilities	
		DACType	Digital-to-Analog converter type	
		ProviderName	Vendor name	
mw_dba.zen_displaydriver	Retrieves the display driver information	Manufacturer	Name of the display driver manufacturer	yes
		Version	Version number of the display driver	
		InstallDate	Install date of the display driver	
		IsShadowed	If True, the display driver is currently being shadowed	
mw_dba.zen_distinguishedname	Retrieves the distinguished name	DistinguishedName	Distinguish name	No
		Tree	eDirectory tree name	
mw_dba.zen_dma	Retrieves the DMA information	Description	Name of the logical device that is currently using this DMA channel	Yes
		DMAChannel	Number of the Direct Memory Access (DMA) channel that a computer uses for transferring data to and from devices quicker than from computers without a DMA channel	
		Availability	Indicates whether Virtual Direct Memory Access (DMA) is supported	
		BurstMode	A data transmission mode in which data is sent faster than normal	
mw_dba.zen_dnsname	Retrieves the DNS name	HostName	DNS name of the inventoried server	No

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_floppy	Retrieves the floppy information	DeviceID	The floppy name representing the floppy	No
		Manufacturer	Vendor name	
		Description	Floppy drive description	
		PhysicalCylinders	Total number of cylinders or tracks on the floppy.	
		PhysicalHeads	Floppy drive R/W heads	
		SectorsPerTrack	Floppy drive sectors per track	
		Capacity	Floppy drive capacity	
mw_dba.zen_inventoryscanner	Retrieves the inventory scanner information	LastScanDate	The date when the Scanner was last scanned. Stored as milliseconds time value so it can be read and displayed in any appropriate date format	Yes
		InventoryServer	Name of the Inventory server to which the scans are sent. It is not the complete DN of the server name	
		Version	Version of the Scanner running on the inventoried server	
		ScanMode	The management technology used by the Scanner, such as WMI or DMI, for scanning the computer system	
		RecentInformation	Latest inventory information	
		generaldictionaryversion	Version of the General dictionary	
		privatedictionaryversion	Version of the Private dictionary	
mw_dba.zen_ipaddress	Retrieves the IP address	Address	The unique address assigned to a computer on an IP Internet	No

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		SubnetMask	The subnet mask of the inventoried server paired with an IP address specifies to an IP router which octets or bits in the IP address are the network ID and which octets or bits are the node ID	
mw_dba.zen_ipxaddress	Retrieves the IPX address	Address	The IPX address of the inventoried server	No
mw_dba.zen_irq	Retrieves the IRQ information	IRQNumber	Number of the Interrupt Request Line (IRQ), from 0 to 15	Yes
		Availability	Availability of the specific IRQ channel	
		TriggerType	IRQ Trigger type	
		Shareable	If True, the system IRQ can be shared across devices	
mw_dba.zen_keyboard	Retrieves the keyboard information	KeyboardLayout	Layout of the keyboard	No
		KeyboardSubtype	Type of the keyboard	
		KeyboardDescription	Description of the keyboard, such as IBM Enhanced 101 or 102 keys	
		NumberOfFunctionKeys	Total number of function keys	
		KeyboardDelay	Delay before the repeat of a key	
		TypematicRate	Rate of processing the keys	
mw_dba.zen_lastlogindetails	Retrieves the last login details	LastUser	User most recently logged into the Primary eDirectory tree through Novell Client when the inventoried server was scanned	No
mw_dba.zen_macaddress	Retrieves the MAC address	MACAddress	Unique node address permanently coded in the network adapter that identifies a specific computer on a network	No
mw_dba.zen_memory	Retrieves the memory information	TotalMemory	Total memory of the inventoried server	No

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_microsoftdomainname	Retrieves the Microsoft domain name	DomainName	Domain name of the inventoried server	No
mw_dba.zen_internalmodem	Retrieves the internal modem information	Name	Identifying information of the modem	No
		Description	Additional information about the modem	
		ProviderName	Name of the vendor	
		DeviceID	Special hexadecimal string identifying the modem type	
mw_dba.zen_monitor	Retrieves the monitor information	DeviceID	Unique ID of a desktop monitor that is attached to an inventoried server. For example, DesktopMonitor1.	No
		ModelID	Unique ID of a model of the monitor. It is a combination of the Manufacturer ID and Product ID. For example, DELA001.	
		MonitorDescription	Description of the monitor.	
		NominalSize	A number representing the diagonal width of the monitor (the distance from one corner of the screen to the opposite corner of the screen) For example, 17” You can customize the scan of the nominal size of the monitor by configuring the HWRules .ini file using the Server Inventory policy.	
		ViewableSize	A number representing the diagonal width of the screen image excluding the black borders around the image’s edge For example, 15.8”	
		ManufacturedDate	Year in which the monitor was manufactured	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		MonitorSerialNumber	Manufacturer's number used to identify a monitor For example, 23DDC24N9067	
		Manufacturer	Name of the monitor's manufacturer For example, DELL Computer Corp	
		Model	Product name of the monitor given by the manufacturer For example, DELL E771a	
mw_dba.zen_motherboard	Retrieves the motherboard information	Description	General description of the motherboard	No
		Manufacturer	Name of the motherboard manufacturer	
		Version	Version of the motherboard	
		NumberOfSlots	The number of expansion slots in the motherboard for adding more memory, graphic capabilities, and support for special devices	
mw_dba.zen_mouse	Retrieves the mouse information	MouseType	Mouse type	Yes
		MouseName	Identifying information of the mouse	
		NumberOfButtons	Number of buttons on the mouse	
		IRQNumber	Interrupt assigned to this device	
mw_dba.zen_mousedriver	Retrieves the mouse driver information	DriverName	Name of the mouse driver	No
		DriverVersion	Version number of the mouse driver	
mw_dba.zen_NetworkAdapter	Retrieves the network adapter information	Caption	Network adapter caption	Yes
		Description	Network adapter description	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		InstallDate	Install date of the network adapter	
		Name	Network adapter name	
		PermanentAddress	Node address stored permanently in the adapter	
		MACAddress	The MAC address stored in the network adapter	
		MaxSpeed	Rate at which the data is transferred over the LAN	
		AdapterType	Type of network adapter, such as FDDI or token ring	
		ProviderName	Name of the provider	
mw_dba.zen_NetworkAdapterDriver	Retrieves the network adapter driver information	Description	Description of the network adapter driver installed on the inventoried server For example, IBM 10/100 Ethernet adapter, EN-2420Px Ethernet adapter	No
		Name	Name of the network adapter driver	
		Version	Version of the network adapter	
mw_dba.zen_parallelport	Retrieves the parallel port information	PortName	The logical name of the input-output device on this parallel port, under this operating environment	Yes
		HasDMASupport	If True, DMA is supported	
		PortAddress	Base I/O address for this parallel port	
		IRQNumber	IRQ number of the parallel port	
mw_dba.zen_powersupply	Retrieves the power supply information	Description	Expanded description of the input voltage capability for this power supply	No
		TotalOutputPower	Attribute value that represents the total output power of the power supply	
mw_dba.zen_processor	Retrieves the processor information	DeviceID	Special hexadecimal string identifying the processor type	Yes

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		Description	Additional information about the processor	
		Role	Type of processor such as central processor, math coprocessor, and others	
		Family	Identification of the processor family such as Pentium II, Pentium III, and others	
		OtherFamilyDescription	Additional description about the Processor Family, such as Pentium Processor with MMX technology	
		UpgradeMethod	The method by which this processor can be upgraded, if upgrades are supported	
		MaxClockSpeed	Maximum clock speed of the processor	
		CurrentClockSpeed	Current clock speed of the processor	
		Stepping	Single-byte code characteristic provided by microprocessor vendors to identify the processor model	
mw_dba.zen_serialport	Retrieves the serial port information	PortName	The logical name of the I/O device on this serial port, under this operating environment	No
		PortAddress	Base input-output address for this serial port	
		IRQNumber	IRQ number of the serial port	
mw_dba.zen_soundadapter	Retrieves the sound adapter information	Name	Label of the multimedia card	No
		Description	Description of the multimedia component for the server	
		ProviderName	Name of the provider	
mw_dba.zen_systemslot	Retrieves the system slot information	SlotDescription	Card currently occupying this slot	No

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		MaxDataWidth	Maximum bus width of cards accepted in the slot	
		ThermalRating	Maximum thermal dissipation of the slot in milliwatts	
mw_dba.zen_unixOS	Retrieves the Unix operating system information	Type	Operating system of the inventoried server	Yes
		Caption	Operating system name	
		CodePage	Language code page of the operating system	
		Version	Version number of the operating system	
		InstallDate	Install date of the operating system	
		VirtualMemorySize	Total number of bytes in the virtual address space of the calling process	
		VisibleMemorySize	Total memory as reported by the operating system	
		ProviderName	Name of the provider	
		KernelVersion	Version number of the operating system	
		SwapSpaceSize	Total swap space size	
mw_dba.zen_windowsOS	Retrieves the Windows operating system information	Type	Operating system of the inventoried server	Yes
		OtherTypeDescription	Additional description of the operating system if available	
		Caption	Operating system name	
		CodePage	Language code page of the operating system	
		Version	Version number of the operating system	
		InstallDate	Install date of the operating system	
		VirtualMemorySize	Total number of bytes in the virtual address space of the calling process	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		VisibleMemorySize	Total memory as reported by the operating system	
		ProviderName	Name of the provider	
mw_dba.zen_NetWareOS	Retrieves the NetWare operating system information	Type	Operating system of the inventoried server	Yes
		Caption	Operating system name	
		CodePage	Language code page of the operating system	
		Version	Version number of the operating system	
		InstallDate	Install date of the operating system	
		VirtualMemorySize	Total number of bytes in the virtual address space of the calling process	
		VisibleMemorySize	Total memory as reported by the operating system	
		SizeStoredInPagingFiles	The total number of KBytes that can be stored in the OperatingSystem's paging files	
		ProviderName	Name of the provider	
		AccountingVersion	NetWare server specific attributes	
		InternetBridgeSupport	NetWare server specific attributes	
		MaxNumberOfConnections	NetWare server specific attributes	
		MaxNumberOfVolumes	NetWare server specific attributes	
		PeakConnectionsUsed	NetWare server specific attributes	
		PrintServerVersion	NetWare server specific attributes	
		QueuingVersion	NetWare server specific attributes	
		RevisionLevel	NetWare server specific attributes	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		SecurityRestrictionLevel	NetWare server specific attributes	
		SFTLevel	NetWare server specific attributes	
		TTSlevel	NetWare server specific attributes	
		VAPVersion	NetWare server specific attributes	
		VirtualConsoleVersion	NetWare server specific attributes	
		InternalNetworkNumber	NetWare server specific attributes	
mw_dba.zen_software	Retrieves the software information	Name	Vendor-defined name of the product represented as a vendor trademark or registered trademark.	Yes
		Vendor	Vendor name of the software	
		Version	User-friendly version of a product. For example, the version for Windows 2000 is 2000 or Major.Minor Version of the Product.	
		ProductID	A unique, 16-character identifier for an installed product. This identifier is available from MSI on Windows The format is ABCD-1234-WXYZ-PQRS	
		InternalVersion	Internal version of a product The format is: *major version.minor version.build.sub build number* or *major version.minor version.build*	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		Language	User-friendly name for the language of this copy of the product	
		FriendlyName	Display name of the software	
		Uninstallstring	The command to invoke for uninstalling this product instance. Currently, this is available in Add-Remove Programs (ARP) and MSI on Windows	
		Supportpack	Installed support pack number of the product	
		SoftwareEdition	Product edition defined by the vendor. For example, Professional	
		LastExecutionTime	Date and time stamp when the product was last executed	
		Frequencyofusage	Number of times the product is used	
		Description	Description of the product	
		InstallationSource	Identifies the file system path where the installation files were stored when installing this product instance. Currently, this is available in ARP and MSI on Windows	
		InformationRepository	Source of scan, which can be, Add-Remove Programs, MSI, Software Dictionary, or PRODUCTS.DAT	
		Category	Product category to which the product belongs For example, Office is a part of the Productivity tools category and Solitaire is a game	
		Helplink	Support web site URL for the product that is available in ARP and MSI	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		PackageGUID	Vendor-defined GUID for a product that is available in MSI	
		Path	Directory path where the product is installed on the inventoried server	
mw_dba.zen_softwaregroup	Retrieves the software group information	Name	Vendor-defined name of the software group represented as a vendor trademark or registered trademark	Yes
		Vendor	Vendor name for the software group	
		Version	User-friendly version of a software group	
		ProductID	A unique, 16-character identifier for an installed product. This identifier is available from MSI on Windows The format is ABCD-1234-WXYZ-PQRS.	
		InternalVersion	Internal version of a product The format is: *major version.minor version.build.sub build number* or *major version.minor version.build*	
		Language	User-friendly name for the language of this copy of the product	
		FriendlyName	Display name of the software	
		Uninstallstring	The command to invoke for uninstalling this product instance. Currently, this is available in Add-Remove Programs (ARP) and MSI on Windows	
		Supportpack	Installed support pack number of the product	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		SoftwareEdition	Product edition defined by the vendor. For example, Professional.	
		LastExecutionTime	Date and time stamp when the product was last executed	
		Frequencyofusage	Number of times the product group is used	
		Description	Description of the product group	
		InstallationSource	Identifies the file system path where the installation files were stored when installing this product instance. Currently, this is available in ARP and MSI on Windows	
		InformationRepository	Source of scan, which can be, Add-Remove Programs, MSI, Software Dictionary, or PRODUCTS.DAT	
		Category	Product category to which the product belongs For example, Office is a part of the Productivity tools category and Solitaire is a game	
		Helplink	Support web site URL for the product that is available in ARP and MSI	
		PackageGUID	Vendor-defined GUID for a product that is available in MSI	
		Path	Directory path where the product is installed on the inventoried server	
mw_dba.zen_softwarepatch	Retrieves the software patch information	productid	Software ID of the software patch	No
		PatchName	Vendor-defined name for the patch	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
mw_dba.zen_antivirus	Retrieves the antivirus product information	Name	Vendor-defined name of the antivirus product represented as a vendor trademark or registered trademark	Yes
		Vendor	Vendor name for the antivirus product	
		Version	User-friendly version of the antivirus product	
		ProductID	A unique, 16-character identifier for an installed antivirus product. This identifier is available from MSI on Windows The format is ABCD-1234-WXYZ-PQRS	
		InternalVersion	Internal version of the antivirus product The format is: *major version.minor version.build.sub build number* or *major version.minor version.build*	
		Language	User-friendly name for the language of this copy of antivirus product	
		FriendlyName	Display name of the antivirus product	
		Uninstallstring	The command to invoke for uninstalling this product instance. Currently, this is available in Add-Remove Programs (ARP) and MSI on Windows	
		Supportpack	Installed support pack number of the antivirus product	
		SoftwareEdition	Antivirus Product edition defined by the vendor	
		LastExecutionTime	Date and time stamp when the antivirus product was last executed	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		Frequencyofusage	Number of times the antivirus product is used	
		Description	Description of the antivirus product	
		InstallationSource	Identifies the file system path where the installation files were stored when installing this antivirus product instance. Currently, this is available in ARP and MSI on Windows	
		InformationRepository	Source of scan, which can be, Add-Remove Programs, MSI, Software Dictionary, or PRODUCTS.DAT	
		DefinitionDate	The date of the virus definition file installed on the computer. Some anti-virus products combine date and version into a single string	
		DefinitionVersion	The vendor-defined version of the virus definition file that has been installed on a computer	
		Category	Product category to which the antivirus product belongs	
		Helplink	Support web site URL for the antivirus product that is available in ARP and MSI	
		PackageGUID	Vendor-defined GUID for the antivirus product that is available in MSI	
		Path	Directory path where the antivirus product is installed on the inventoried server	
mw_dba.zen_dictionaryfile	Retrieves the ZENworks software dictionary file information	fileid	Dictionary File ID	Yes
		directoryid	Directory ID	
		FileName	Filename of the dictionary file	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		Directory	Directory name in which the dictionary file is stored	
		FileVersion	Dictionary file version	
		FileSize	Dictionary file size	
		LastModified	Last modified date of the dictionary file	
		InternalName	Internal name	
		ProductVersion	The version of the product represented by this file	
		Company	Vendor name	
		ProductName	The product which this file represents	
		Language	User-friendly name for the language of this copy of the file	
		SoftwareDictionaryID	ID of the file as represented in the General software dictionary	
mw_dba.zen_excludedfile	Retrieves the excluded file information	fileid	Excluded file ID	Yes
		directoryid	Directory ID	
		FileName	Filename of the excluded file	
		Directory	Directory name in which the excluded file is stored	
		FileVersion	Excluded file version	
		FileSize	Excluded file size	
		LastModified	Last modified date of the excluded file	
		InternalName	Internal name	
		ProductVersion	The version of the product represented by this file	
		Company	Vendor name	
		ProductName	The product which this file represent	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		Language	User-friendly name for the language of this copy of the file	
mw_dba.zen_locktable	Retrieves the lock table information	ComputerName	Computer name	Yes
		LastScanTime	The date when the Scanner was last scanned. Stored as milliseconds time value so it can be read and displayed in any appropriate date format	
		RecentInformation	Latest information	
mw_dba.zen_removabledisk	Retrieves the removable disk information	Manufacturer	Vendor name for the removable disk	No
		Description	Description of the removable disk	
		PhysicalCylinders	Total number of cylinders or tracks on the disk	
		PhysicalHeads	Number of heads	
		SectorsPerTrack	Number of sectors per track	
		Capacity	Total size	
mw_dba.zen_fixeddisk	Retrieves the fixed disk information	Manufacturer	Vendor name of the fixed disk	No
		Description	Description of the fixed disk	
		PhysicalCylinders	Total number of cylinders or tracks on the disk	
		PhysicalHeads	Number of heads	
		SectorsPerTrack	Number of sectors per track	
		Capacity	Total size	
mw_dba.zen_WindowsLocalFileSystem	Retrieves the Windows local file system information	Name	Windows local file system name	No
		FileSystemSize	Windows local file system size	
		AvailableSpace	Windows local file system space	
		FileSystemType	Windows local file system type	

Inventory View Name	Functionality	Attributes	Description	Is the view Localized?
		Caption	Windows local file system caption	
		DeviceID	Device ID	
		VolumeSerialNumber	Windows local file system volume number	
mw_dba.zen_NetWareLocalFileSystem	Retrieves the NetWare local file system information	Name	NetWare local file system name	No
		FileSystemSize	NetWare local file system size	
		AvailableSpace	NetWare local file system available space	
		FileSystemType	NetWare local file system type	
		Caption	NetWare local file system caption	
		DeviceID	Device ID	
		VolumeSerialNumber	NetWare local file volume serial number	
mw_dba.zen_LinuxLocalFileSystem	Retrieves the Linux local file system information	Name	Linux local file system name	No
		FileSystemSize	Linux local file system size	
		AvailableSpace	Linux local file system available space	
		FileSystemType	Linux local file system type	
		Caption	Linux local file system caption	
		DeviceID	Device ID	
		VolumeSerialNumber	Linux local file system volume serial number	

How to Use the Inventory Views

You can use the Inventory views along with SQL statements, and execute the SQL statements from the Inventory database prompt or in any third-party database front-end application.

IMPORTANT: You must not edit the Inventory views that ship with ZENworks 6.5 Server Management SP1 or later.

Examples:

- To retrieve all the processor information:

  ```
  select * from mw_dba.zen_processor
  ```

- To retrieve specific processor information:

  ```
  select DeviceID, Description, Role, Family, OtherFamilyDescription,
  UpgradeMethod, MaxClockSpeed, CurrentClockSpeed from mw_dba.zen_processor
  ```

- To retrieve all software information:

  ```
  select * from mw_dba.zen_software
  ```

- To retrieve software information along with its suite details:

  ```
  select soft.name, softsuite.name from mw_dba.zen_software soft,
  mw_dba.zen_softwaregroup softsuite where soft.name = 'ZENworks Desktop
  Management Inventory Server' and
  soft.parentinstanceid=softsuite.pinstanceid;
  ```

- To retrieve software patch information:

  ```
  select suite.name, patchname from mw_dba.zen_softwaregroup suite,
  mw_dba.zen_softwarepatch patch where suite.pinstanceid=patch.pinstanceid
  ```

- To retrieve software suite patch information:

  ```
  select soft.name, patchname from mw_dba.zen_software soft,
  mw_dba.zen_softwarepatch patch where soft.pinstanceid=patch.pinstanceid;
  ```

- To retrieve anti-virus software information:

  ```
  select * from mw_dba.zen_antivirus
  ```

18 Monitoring Server Inventory Using Status Logs

Novell® ZENworks® 6.5 Server Inventory lets you track whether the scan or the roll-up of information is successful by viewing the log files for scan status, roll-up status and Inventory server status.

The scan status of the inventoried server is reported through local log files.

The inventory components report the status of the Inventory server and roll-up of scan information in Novell eDirectory™.

For example, when you view the status logs, you can determine whether the processing of the scan files was successful or if there were any errors while scanning the server or at the time of roll-up.

You can view the following status information:

Viewing the Scan Status of an Inventoried Server

The Inventory Agent reports status information and errors in the invagent.log file. This log file is stored in the sys:\etc directory on NetWare® servers and in the temp directory or the windows\temp directory on Windows servers.

The native scanner reports status information and errors in the invnatve.log file. This log file is stored in the sys:\etc directory on NetWare servers and in the temp directory or the windows\temp directory on Windows servers.

The Inventory Policy Enforcer writes the status of the current invocation by the policy engine into the invagentpolicyenforcer.log file. You can set the debug flag in the file invsetup.ini located in sys:\system or %systemroot%.

In the forceDebug=true mode, the Inventory Agent writes the status of the .str file transfer into the invagentstrtransfer.log file. This file will be located in the sys:\system\invscan directory on NetWare servers and in the *%sytem_drive%*\invscan directory on Windows servers.

Viewing the Roll-Up History of the Inventory Server

The Roll-Up Status reports the status of the roll-up information from the Inventory server that initiated the roll-up of information. For example, if your inventory setup consists of a Leaf Server which initiates the roll-up of information to the next-level Root Server, the Roll-Up log displays the roll-up history of the Leaf Server.

The inventory components of the Inventory server (Sender, Receiver, and Storer) write the scan information in the Roll-Up Status. For example, you view the Roll-Up log to determine whether there were any errors during roll-up of inventory information from the Inventory server. This log also displays the most recent roll-up time of the inventory information that was stored in the database on the topmost level server (Root Server). This log displays the history of the ten previous roll-up sessions done from the Inventory server.

The following table lists the details of the log:

Status Information	Details
Roll-Up Start Time	Displays the date and time of the roll-up.
Message	Displays the message reported by the inventory component while moving the inventory information across the Inventory servers.

You can export the file as a .csv or tab-delimited file.

To invoke the Roll-Up Status window:

1. In ConsoleOne®, right-click the Inventory Service object (Inventory Service_*server_name*), from which the roll-up is done, click Properties, click Status Report tab, then click Roll-Up Status.

Viewing the Status of Inventory Components on an Inventory Server

The Server Status window reports the status of the Inventory server components on the selected Inventory server. You can view the Inventory server Status log for any Inventory Service object. For example, you can determine whether the Sender sent the files to the Receiver or whether the Storer was able to establish the connection with the database successfully. The Server Status window displays the details of the ten latest status messages logged by the Inventory server components.

If the Inventory server components (Sender, Receiver, Selector, Storer, Scan Collector, Service Manager, or Roll-Up Scheduler) are not up and running on the Inventory server, the status of the Inventory server displays the information.

The following table lists the details of the log:

Status Information	Details
Time of Log	Displays the date and time when the message was reported by the inventory components.
Source	Displays the inventory component that has logged the status message.

Status Information	Details
Message Type	Displays the severity of the message.
Message	Displays the message reported by the inventory components.

You can export the log file as a .csv or tab-delimited file.

To view the Server Status window:

1. In ConsoleOne, right-click the Inventory Service object (Inventory Service_*server_name*), then click Properties, click Status Report, then click Server Status.

Viewing the Status of the Last Scan on the Inventoried Server

On NetWare, Windows servers, the invagent.log and the invnatve.log files will store the details and last execution status of the Inventory scan.

Viewing the Roll-Up Log for the Inventory Servers

The Roll-Up log reports the status of the latest roll-up from the Inventory Service objects in the container. For example, you view the Roll-Up log to determine whether the latest roll-up of information from the Roll-Up server for the Inventory Service object was successful. The inventory components (Sender, Receiver, and Storer) write the roll-up information in the Roll-Up log. You can also choose to display error, warning, and informational status messages of the Intermediate servers.

The following table lists the details of the log:

Status Information	Details
Roll-Up Initiated From	Displays the DN of the Intermediate Server that initiated the roll-up.
Roll-Up Start Time	Displays the date and time the roll-up of information was initiated.
Source	Displays the inventory component that logs the status.
Message Type	Displays the severity of the message.
Message	Displays the message reported by the inventory components while scanning the inventoried server.

You can export the log as a .csv or tab-delimited file.

To invoke the Roll-Up Log window:

1. In ConsoleOne, click the container that contains the Inventory Service object (Inventory Service_*server_name*), click Tools, click ZENworks Inventory, then click Roll-Up Log.
2. Click the severity type of the messages you want to view, then click OK.

Exporting the Inventory Status Log Files

You can store the details of the log files as Comma-Separated-Value reports or as a tab-delimited file.

To save the log as a file:

1. In ConsoleOne, open the Status window.
2. Click Export.
3. Select the file type, and specify the filename.
4. Click OK.

Overview of Status Logs and Scan Logs

The following table lists the status logs and scan logs:

Status/Scan Log	Inventory Components that Log the Status	Details of the Log	How to View the Log File
Inventoried Server Scan Log	Scan program, Policy Enforcer	Format module name, time stamp, status code and status message	Available locally on the inventoried server
Roll-Up Log	Sender, Receiver, Storer	Roll-up initiated from, roll-up start time, inventory component, message type, status message	Click the container for the Inventory Service object, click Tools, click ZENworks Inventory, then click Roll-Up Log
Invagent.log	Scan program, Inventory Agent	Format module name, time stamp, status code and status message	Opens in any text editor
Invnatve.log	Scan program	Format module name, time stamp, status code and status message	Opens in any text editor
Invagentpolicyenforcer.log	Policy Enforcer	Time of log, error type, description, severity and state	Opens in any text editor
Invagentstrtransfer.log (created in the debug mode)	Inventory Agent	Time of log, error type, description, severity and state	Opens in any text editor
Status of Inventory components on Server	Sender, Receiver, Scan Collector, Selector, Storer, Service Manager, Roll-Up Scheduler	Time of log, source, message type, message	In ConsoleOne, right-click the Inventory Service object, click Properties, click Status Report, then click Server Status
Roll-Up Status	Sender, Receiver, Storer	Roll up start time, message	In ConsoleOne, right-click the Inventory Service object, click Properties, click Status Report , then click Roll-Up Status

Viewing the Status Log in XML Format

All inventory components log the status messages in a log file maintained in XML (Extensible Markup Language) format. Unlike the status logs that contain a history of the ten latest status messages, the status XML log stores all status messages.

The log file contains the following information:

- Inventory module name
- Date and time of status logging
- Severity of the message
- Message text and status message number
- DN name, if the inventory module is associated with a particular DN object in eDirectory
- Product-specific details of the module

The format of the log file is as follows:

```
?xml version="1.0" encoding="UTF-8"?>

?xml stylesheet type="text/xsl" href="inventorylog.xsl"?

<message_log>

 <message_entry>

   <module_name>Scanner</module_name>

   <severity>Critical</severity>

   <date_time>8/3/00 12:49 PM</date_time>

    <message_tag>unable to create scan data files
    </message_tag>

    <dn_name>Inv_server</dn_name>

  </message_entry>

  </module_name>Storer</module_name>

    <severity>Critical</severity>

    <date_time>8/3/00 12:49 PM</date_time>

    <message_tag>unable to update the database</message_tag>

    <dn_name>Inv_server</dn_name>

</message_entry>

..

</message_log>
```

A sample style sheet and Document Type Declaration (DTD) file are located in *inventory_installation_directory*\inv\server\xmllog on the Inventory server.

The inventorylog.xml log file is located in the *inventory_installation_directory* \inv\server\xmllog directory on NetWare and Windows Inventory servers.

By default, the maximum size of the log file is 100 KB. To modify the maximum size of the log file, edit the inventorylog.ini file. On NetWare and Windows Inventory servers, this file is in the *inventory_installation_directory*\inv\server\xmllog directory.

The contents of inventorylog.ini are as follows:

```
max_file_size=100 KB
```

Modify the MAX_FILE_SIZE parameter, if required.

If the file size exceeds the value specified in the MAX_FILE_SIZE parameter, the file is archived as *filename*_old.xml. The latest messages will be in the current log file.

To view the log data file, use a third-party XML browser.

H Performance Tips

This section provides information on the system and database parameters that you need to tune to obtain improved performance for the Server Inventory component of Novell® ZENworks® 6.5 Server Management. Specific tuning tips are provided for working with Inventory Reports, Database Export, and Query.

In addition to reviewing this information, we recommend that you refer to vendor documentation or other related articles regarding performance tuning and database tuning available on the Internet

This chapter contains the following sections:

Database Parameter Tuning Tips

Sybase in the NetWare and Windows Environment

- We recommend you to set the database cache size as follows by configuring the *-c* parameter in the Sybase startup:

Inventoried Servers in the Database (thousands)	Total Memory of the System	Sybase Cache Memory
less than 1000	384 MB	128 MB
1 - 5	512 MB	128 MB
5 - 10	512 MB - 768 MB	128 MB - 256 MB
10 - 25	768 MB - 1 GB	256 MB - 400 MB
greater than 25	1 - 2 GB	30 - 40% of RAM

- If you have more than 5,000 inventoried servers, we recommend that you use multi processors for servers hosting the database and span the data files.
- If you have more than 10,000 inventoried servers, we recommend that you use a dedicated server for the database.

- Ensure that the drives in which the database files are located have sufficient free disk space for storing the temporary files generated during the operations of Inventory ConsoleOne utilities.
- If the Storer is taking significant time to store the inventory information in the following scenarios, you can run the Sybindex utility to improve the Storer performance:
  - Many Inventory agents are simultaneously upgraded to ZENworks 6.5 and subsequently, all these agents send the full scans for the time to the Inventory server.
  - The administrator manually triggers full scan from the Inventory Service object resulting in all Inventory agents send the full scan to the Inventory server.
  - When the Inventory database is either re-installed or changed, the administrator manually triggers full scan from the Inventory Service object resulting in all Inventory agents sending the full scan to the Inventory server.

    Before running the Sybindex utility, ensure that the Sybase Inventory database is up and running, and then stop the Storer (at the console prompt, enter **`stopser storer`**).

    To run the Sybindex utility:

    On a NetWare server: At the server console prompt, enter **`sybindex`**.

    On a Windows server: At the server console prompt, go to *inventory_server_installation_path*\zenworks\inv\server\wminv\bin and enter **`sybindex`**.

    After the utility runs, restart Storer (at the console prompt, enter **`startser storer`**).

To change the database cache size on the database server on NetWare database server:

1. Stop the Inventory service. For more information, see "Starting and Stopping the Inventory Service" on page 454.
2. Close all connections to the Inventory database.
3. Quit the Sybase server.
4. Open the mgmtdbs.ncf file in the sys:\system directory.
5. Modify the *-c* parameter.

   For example, -c 64M sets the cache size to 64 MB.
6. Save the file.
7. On the server console, load the Inventory database. Enter **`MGMTDBS.`**
8. Start the Inventory service. For more information, see "Starting and Stopping the Inventory Service" on page 454.

To change the database cache size on a Windows database server:

1. Stop the Inventory service. For more information, see "Starting and Stopping the Inventory Service" on page 454.
2. Stop the Sybase service.

   On Windows 2000/2003, in the Control Panel, double-click Administrative Tools, double-click Services, select Novell Database - Sybase, then click Stop.

3 On the database server, run the ntdbconfig.exe file from the *inventory_database_installation_path*\zenworks\database\dbengine directory.

   Ntdbconfig.exe is a ZENworks database configuration utility for the ZENworks database using Sybase on Windows servers. This utility enables you to reconfigure the Sybase service. For the list of parameters recommended by Sybase, see "Understanding the Sybase Database Startup Parameters" on page 469.

4 Modify the *-c* parameter.

5 Click OK.

6 Restart the Sybase service.

   On Windows 2000/2003, in the Control Panel, double-click Administrative Tools, double-click Services, select Novell Database - Sybase, then click Start.

7 Start the Inventory service. For more information, see "Starting and Stopping the Inventory Service" on page 454.

For more information on Performance tips, see "Database Parameter Tuning Tips" on page 683.

Oracle in the Windows Environment

- The following table lists the memory recommendations for Windows platforms.

Inventoried Servers in the Database (thousands)	Total Memory of the System	Oracle SGA Memory
less than 1	512 MB	128 MB
1 - 5	768 MB	256 MB
5 - 10	1 GB	400 MB
10 - 100	1 GB - 2 GB	40% of the total memory

- Stop unnecessary services and applications running on the server to enable a background service such as Oracle server to run.
- Ensure that the drives in which the database files are located have sufficient free disk space for storing the temporary files generated during the operations of Inventory ConsoleOne utilities.
- We recommend that you use a dedicated server to host the Oracle database.
- Span the data files across multiple physical disks if you have more than 10,000 inventoried servers.
- Set the virtual memory value between 2 - 4 times the RAM.
- We recommend that you use multi processors for servers hosting the database.
- For Windows platforms, refer to the Oracle performance tuning documentation and other general recommendations that are listed in the "References" on page 688 section.
  - Reduce the priority of the foreground application.
  - Reduce the file cache value and maximize data for network applications.

- Modify the init.ora file for specific organizational requirements.

  For example, to obtain about 260 MB of Oracle SGA with db_block_size=4096, modify the init.ora file with the following values:

  db_block_buffers = 50000
  shared_pool_size = 32768000
  sort_area_size = 10000000

- Invoke and append lines to the _start.sql file. The _start.sql file is invoked by the mgmtdbo.ncf or the mgmtdbo.bat file when you start the Inventory database instance. Append the following lines to the existing _start.sql file:

  ```
  connect mw_dba;

  alter table cim.t$product cache;
  ```

- If you run Inventory database on Oracle9i, you can set db_cache_size instead of db_block_buffers * db_block_size.
- If the Storer is taking significant time to store the inventory information during the full scan, run the following script to create an additional index on InstalledFile table to improve the storing time:

  ```
  create index i$installedfile$compid on mw_dba.installedfile(computerid)
  tablespace cim5 pctfree 0;
  ```

- Refer to the Oracle Administration guide or Performance guide for more information.

Optimizing the Performance of the Oracle Database

If you have an Inventory database on Oracle, you can improve the performance of the database when you generate the inventory reports or query the database.

You use the database buffer cache to store the most recently used data blocks. The database cache is determined as db_block_buffers * db_block_size. These parameters are specified in the zenworks\database\init.ora file on the database server.

DB_BLOCK_BUFFERS specifies the number of database buffers. DB_BLOCK_SIZE specifies the size of each database buffer in bytes.

The size of each buffer in the buffer cache is equal to the size of the data block.

If there is additional memory, you configure the database cache size by increasing the DB_BLOCK_BUFFERS parameter in the init.ora file. If you run Inventory database on Oracle9i, you can set set db_cache_size instead of db_block_buffers * db_block_size

For more information for Performing tips, see “Database Parameter Tuning Tips” on page 683.

MS SQL in the Windows Environment

- We recommend that you use a dedicated server for MS SQL.
- On MS SQL server, ensure that the tempdb system database is located on the drive having sufficient disk space.
- Boost the MS SQL server priority.
- Enable optimization for background services.

- Use the configuration in the following table:

Inventoried Servers in the Database (thousands)	Total Memory of the System	MS SQL Cache Memory	Processor Speed
less than 10	512 MB	256 MB	Pentium III: 450 M Hz
10 - 20	512 MB - 1 GB	256 MB - 384 MB	Pentium 4: 1.8 G Hz
20 - 50	1 GB - 1.5 GB	512 MB - 768 MB	Pentium 4: 1.8 G Hz

- Span the data files across the multiple physical disks if you have more than 5,000 inventoried servers.
- We recommend that you use multi processors for servers hosting the database.
- For additional tips on MS SQL, refer to the MS SQL Server documentation (http://www.ms-sql-server-performance.com).

Performance Tips for the Inventory ConsoleOne Utilities

This section discusses the performance tips for the Inventory ConsoleOne utilities:

- "Inventory Reports Performance Tips" on page 687
- "Inventory Data Export Performance Tips" on page 687
- "Inventory Query Performance Tips" on page 688

NOTE: If the Inventory ConsoleOne utilities are retrieving data from a large database, you must stop the Storer service before running the utilities. This improves the performance of the ConsoleOne utilities.

Inventory Reports Performance Tips

If you have more than 1000 inventoried servers in your database, listing all of the subreports consumes time. We recommend that you specify the list of subreports. By doing so, the general performance of the reports is improved.

Inventory Data Export Performance Tips

- To maximize the performance of Inventory Data Export, you need to enable the filter condition in DBExport. Based on the query you specify, DBExport will export only selected software.
- During export, deselect the attributes that you do not want to use. To do this, use the DBExport and the Required Attributes Only option.
- Perform the software export separately. This will greatly improve the performance of the Non-Software Export function.

Inventory Query Performance Tips

- Specify queries using the AND condition in multiple groups to increase performance.
- Split a complex query with several logical operators into multiple groups separated by a logical operator.
- If you want to use a complex query, increase the database cache size. For more information on tuning databases, see “Database Parameter Tuning Tips” on page 683.
- Save fast, narrowed-down queries for future use.
- Do not invoke the Inventory Query by connecting to a database over a slow link.
- If a complex query takes more than 10 minutes to execute over a fast link, you probably do not have any inventoried servers that match the query you specified. The following message will be displayed:

  ```
  No Computer system matched the query
  ```

  Close the Result window, narrow your input query and retry. Repeat the process of narrowing your query until you locate your inventoried servers.
- For optimal performance, we recommend that you do not use more than four groups and not more than three logical operators separating the four groups in your query.
- If you know the exact logical string, avoid using the MATCHES operator. The MATCHES operator searches the database for a result based on the pattern you specify. This will result in performance degradation.
- If you want to check for a particular inventory component not stored in the Inventory database, use the (ISNULL) operator instead of a query with a regular attribute.

References

For additional information on performance tuning tips, refer to the following documentation for specific components:

- MS SQL performance information (http://www.sql-server-performance.com)
- Oracle9i Database and Performance guide and reference
- Oracle9i Database Administrator’s guide

Hardware Information Collected by the Inventory Scanners

This section provides information on the following topics:

Hardware Information Collected on NetWare Inventoried Servers

The Inventory Agent collects the following hardware information on the NetWare inventoried servers:

Scan Data	SNMP Details	SMBIOS Details
System.Type	SNMP v2.0 RFC1213.MIB	Not applicable
System.MachineName	SNMP v2.0 RFC1213.MIB	Not applicable
System.AssetTag	Not applicable	SMBIOS v2.3 Type 3 structure
System.Model	Not applicable	SMBIOS v2.3 Type 1 structure
System.ModelNumber	Not applicable	SMBIOS v2.3 Type 3 structure
System.SystemIdentifier	Not applicable	Not applicable
System.ManagementTechnology	Not applicable	Not applicable
System.DNName	Not applicable	Not applicable
System.TreeName	Not applicable	Not applicable
NetworkAdpater.MACAddress	SNMP v2.0 RFC1213.MIB	Not applicable
IP.Address	SNMP v2.0 RFC1213.MIB	Not applicable
IP.Subnet (Subnet Mask)	SNMP v2.0 RFC1213.MIB	Not applicable
NetworkAdapter.MACAddress	Not applicable	Not applicable
IPX.Adress	SNMP v2.0 IPX.MIB	Not applicable
NetworkAdapter.MACAddress	SNMP v2.0 IPX.MIB	Not applicable
DNS.HostName	Not applicable	Not applicable
NetworkAdapter.Speed	SNMP v2.0 RFC1213.MIB	Not applicable

Scan Data	SNMP Details	SMBIOS Details
NetworkAdapter.Name	SNMP v2.0 RFC1213.MIB	Not applicable
NetworkAdapter.PermAddress	Not applicable	Not applicable
NetworkAdapter.AdapterType	SNMP v2.0 RFC1213.MIB	Not applicable
NetworkAdapter.ProviderName	SNMP v2.0 RFC1213.MIB	Not applicable
NetworkAdapter.DriverDescription	SNMP v2.0 RFC1514.MIB	Not applicable
NetworkAdapter.DriverName	SNMP v2.0 RFC1514.MIB	Not applicable
NetworkAdapter.DriverVersion	SNMP v2.0 RFC1514.MIB	Not applicable
Zenworks_ZENNetworkAdapter---offset	SNMP v2.0 RFC1514.MIB	Not applicable
Processor.stepping	Not applicable	Not applicable
Processor.DeviceID	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.Family	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.OtherFamily	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.MaxClockSpeed	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.CurrentClockSpeed	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.Role	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.UpgradeMethod	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.Description	Not applicable	SMBIOS v2.3 Type 4 structure
Processor.Name	Not applicable	SMBIOS v2.3 Type 4 structure
BIOS.Manufacturer	Not applicable	SMBIOS v2.3 Type 0 structure
BIOS.BIOSDate	Not applicable	SMBIOS v2.3 Type 0 structure
BIOS.BIOSIDBytes	Not applicable	Not applicable
BIOS.Caption	Not applicable	Not applicable
BIOS.SerialNumber	Not applicable	Not applicable
BIOS.Version	Not applicable	SMBIOS v2.3 Type 0 structure
BIOS.PrimaryBIOS	Not applicable	Not applicable
BIOS.Size	Not applicable	Not applicable
Bus.Type	SNMP v2.0 RFC1514.MIB	Not applicable
Bus.Name	Not applicable	Not applicable
Bus.Description	SNMP v2.0 RFC1514.MIB	Not applicable
Bus.Version	Not applicable	Not applicable

Scan Data	SNMP Details	SMBIOS Details
Monitor.NumberOfColorPlanes	Not applicable	Not applicable
Monitor.HorizontalResolution	Not applicable	Not applicable
Monitor.VerticalResolution	Not applicable	Not applicable
Monitor.DisplayType	Not applicable	Not applicable
Monitor.MemoryType	Not applicable	Not applicable
Monitor.MaxMemorySupported	Not applicable	Not applicable
Monitor.Bitsperpixel	Not applicable	Not applicable
Monitor.ControllerDescription	Not applicable	SMBIOS v2.3 Type 10 structure
Monitor.MaxRefreshrate	Not applicable	Not applicable
Monitor.MinRefreshrate	Not applicable	Not applicable
Mointor.DACType	Not applicable	Not applicable
Monitor.ChipSet	Not applicable	Not applicable
Monitor.ProviderName	Not applicable	Not applicable
Monitor.VideoBIOSManufacturer	Not applicable	Not applicable
Monitor.VideoBIOSVersion	Not applicable	Not applicable
Monitor.VideoBIOSReleaseDate	Not applicable	Not applicable
Monitor.VideoBIOS.IsShadowed	Not applicable	Not applicable
ParallelPort.Name	Not applicable	SMBIOS v2.3 Type 8 structure
ParallelPort.DMASupport	Not applicable	Not applicable
ParallelPort.Address	Not applicable	Not applicable
ParallelPort.IRQ	Not applicable	Not applicable
SerialPort.Name	Not applicable	Not applicable
SerialPort.Address	Not applicable	SMBIOS v2.3 Type 8 structure
SerialPort.IRQ	Not applicable	Not applicable
CDROMDrive.DeviceID(*)	Not applicable	Not applicable
CDROMDrive.Manufacture	Not applicable	Not applicable
CDROMDrive.Description	SNMP v2.0 RFC1514.MIB	Not applicable
CDROMDrive.Caption	SNMP v2.0 RFC1514.MIB	Not applicable
HardDrive.Media Type	SNMP v2.0 RFC1514.MIB	Not applicable
HardDrive.Vendor	Not applicable	Not applicable

Scan Data	SNMP Details	SMBIOS Details
HardDisk.Description	SNMP v2.0 RFC1514.MIB	Not applicable
HardDisk.Cylinders	Not applicable	Not applicable
HardDisk.Heads	Not applicable	Not applicable
HardDisk.Sectors	Not applicable	Not applicable
HardDisk.Capacity	SNMP v2.0 RFC1514.MIB	Not applicable
FileSystem.Name	Not applicable	Not applicable
InventoryScanner.Version	Not applicable	Not applicable
InventoryScanner.LastScanDate	Not applicable	Not applicable
InventoryScanner.InventoryServer	Not applicable	Not applicable
InventoryScanner.ScanMode	Not applicable	Not applicable
SoundCard.Description	Not applicable	SMBIOS v2.3 Type 10 structure
SoundCard.Name	Not applicable	Not applicable
SoundCard.Manufacturer	Not applicable	Not applicable
Cache.Level	Not applicable	Not applicable
Cache.WritePolicy	Not applicable	Not applicable
Cache.ErrorCorrection	Not applicable	SMBIOS v2.3 Type 7 structure
Cache.Type	Not applicable	SMBIOS v2.3 Type 7 structure
Cache.LineSize	Not applicable	Not applicable
Cache.ReplacementPolicy	Not applicable	Not applicable
Cache.ReadPolicy	Not applicable	Not applicable
Cache.Associativity	Not applicable	SMBIOS v2.3 Type 7 structure
Cache.Speed	Not applicable	SMBIOS v2.3 Type 7 structure
Cache.Size	Not applicable	Not applicable
UCS.DNName	Not applicable	Not applicable
UCS.PrimaryOwnerContact	Not applicable	Not applicable
UCS.PrimaryOwnerName	Not applicable	Not applicable
Slot.Description	Not applicable	SMBIOS v2.3 Type 9 structure
Slot.MaxDataWidth	Not applicable	SMBIOS v2.3 Type 9 structure
Slot.ThermalRating	Not applicable	Not applicable
LogicalDrive.Name	Not applicable	Not applicable

Scan Data	SNMP Details	SMBIOS Details
LogicalDrive.DeviceID	Not applicable	Not applicable
LogicalDrive.VolumeSerialNumber	Not applicable	Not applicable
FileSystem.Name	Not applicable	Not applicable
FileSystem.Type	Not applicable	Not applicable
FileSystem.TotalSize	Not applicable	Not applicable
FileSystem.FreeSpace	Not applicable	Not applicable
FileSystem.DeviceID	Not applicable	Not applicable
Operating System.OSType	Not applicable	Not applicable
OperatingSystem.Version	Not applicable	Not applicable
OperatingSystem.Codepage	Not applicable	Not applicable
OperatingSystem.InstallDate	Not applicable	Not applicable
OperatingSystem.SizeStoredInPagingFiles	Not applicable	Not applicable
OperatingSystem.Caption	Not applicable	Not applicable
OperatingSystem.TotalVisibleMemorySize	Not applicable	Not applicable
OperatingSystem.Role	Not applicable	Not applicable
NetWareOperatingSystem.AccountingVersion	Not applicable	Not applicable
NetWareOperatingSystem.InternetBridgeSupport	Not applicable	Not applicable
NetWareOperatingSystem.MaxNumberOfConnections	Not applicable	Not applicable
NetWareOperatingSystem.PeakConnectionsUsed	Not applicable	Not applicable
NetWareOperatingSystem.PrintServerVersion	Not applicable	Not applicable
NetWareOperatingSystem.QueuingVersion	Not applicable	Not applicable
NetWareOperatingSystem.RevisionLevel	Not applicable	Not applicable
NetWareOperatingSystem.SecurityRevisionLevel	Not applicable	Not applicable
NetWareOperatingSystem.SFTLevel	Not applicable	Not applicable
NetWareOperatingSystem.TTSLevel	Not applicable	Not applicable
NetWareOperatingSystem.VAPVersion	Not applicable	Not applicable
NetWareOperatingSystem.VirtualConsoleVersion	Not applicable	Not applicable
NetWareOperatingSystem.InternalNetworkNumber	Not applicable	Not applicable

Hardware Information Collected on Windows Inventoried Servers

The Inventory Agent collect the following hardware information on the Windows inventoried servers:

Scan Data	DMI Class and Attribute	WMI Class and Attribute
System.Manufacturer	DMTF|Component| 1	Win32_ComputerSystemProduct.Vendor
System.MachineName	Not applicable	Win32_ComputerSystem.Caption
System.AssetTag	DMTF|System Enclosure| 001.2	Not applicable
System.Model	DMTF|Component|2	Win32_ComputerSystemProduct.Name
System.ModelNumber	Not applicable	Not applicable
System.SystemIdentifier(GUID)	Not applicable	Not applicable
System.SerialNumber	DMTF|Component|3	Win32_ComputerSystemProduct.IdentifyingNumber
System.Tag	Not applicable	Not applicable
System.ManagementTechnology	Not applicable	Not applicable
eDirectory.DNName	Not applicable	Not applicable
eDirectory.TreeName	Not applicable	Not applicable
NetworkAdapter.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MACAddress (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)
IP.Address	Not applicable	Win32_NetworkAdapterConfiguration.IPAddress (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)
IP.Subnet (Subnet Mask)	Not applicable	Win32_NetworkAdapterConfiguration.IPSubnet (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)
NetworkAdapter.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MACAddress (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)
IPX.Address	Not applicable	Win32_NetworkAdapterConfiguration.IPXAddress (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)
NetworkAdapter.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MACAddress (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)

Scan Data	DMI Class and Attribute	WMI Class and Attribute
DNS.HostName	Not applicable	Win32_NetworkAdapterConfiguration.DNSHostName + DNSDomain (Only on Windows 2000; get it through association with Win32_NetworkAdapterSetting)
Modem.Description	Not applicable	Win32_POTSModem.Description
Modem.Name	Not applicable	Win32_POTSModem.Name
Modem.Vendor	Not applicable	Not applicable
Modem.DeviceID	Not applicable	Win32_POTSModem.DeviceID
NetworkAdapter.DriverVersion	DMTF|Network Adapter Driver |001.Driver Software Version	Not applicable
Login.CurrentLoggedinUser	Not applicable	Not applicable
Login.LastLoggedin User	Not applicable	Not applicable
Login.DomainName	Not applicable	Win32_ComputerSystem.Domain
NWClient.Version	Not applicable	Not applicable
Processor.stepping	Not applicable	CIM_Processor.Stepping
Processor.DeviceID	Not applicable	CIM_Processor.DeviceID
Processor.Family	DMTF|Processor|004.3	CIM_Processor.Family
Processor.OtherFamily	Not applicable	CIM_Processor.OtherFamilyDescription
Processor.MaxClockSpeed	DMTF|Processor|004.5	CIM_Processor.MaxClockSpeed
Processor.CurrentClockSpeed	DMTF|Processor|004.6	CIM_Processor.CurrentClockSpeed
Processor.Role	DMTF|Processor|004.2	CIM_Processor.ProcessorType
Processor.Upgrade	DMTF|Processor|004.7	CIM_Processor.UpgradeMethod
Processor.Description	DMTF|Processor|004.4	CIM_Processor.Description
Processor.Name	Enum equivalent of DMTF|Processor|004.3	CIM_Processor.Name
BIOS.Manufacturer	DMTF|SystemBIOS|001.2	Win32_BIOS.Manufacturer
BIOS.BIOSDate	Not applicable	Win32_BIOS.InstallDate
BIOS.BIOSIDBytes	Not applicable	Not applicable
BIOS.Copyright	Not applicable	Win32_BIOS.Caption
BIOS.SerialNumber	Not applicable	Win32_BIOS.SerialNumber
BIOS.BIOSType	DMTF|SystemBIOS|001.3	Win32_BIOS.SMBIOSBIOSVersion

Scan Data	DMI Class and Attribute	WMI Class and Attribute
BIOS.PrimaryBIOS	DMTF|SystemBIOS|001.9	Win32_BIOS.PrimaryBIOS
BIOS.Size	DMTF|SystemBIOS|001.4	Not applicable
Bus.Type	Not applicable	Win32_Bus.BusType
Bus.Name	Not applicable	Win32_Bus.Name
Bus.Description	Not applicable	Win32_Bus.Descritpion
Bus.Version	Not applicable	Not applicable
Bus.DeviceID	Not applicable	Win32_Bus.DeviceID
IRQ.Number	DMTF|IRQ|002.IRQNumber	CIM_IRQ.IRQNumber
IRQ.Availability	DMTF|IRQ|002.Availability	CIM_IRQ.Availability
IRQ.TriggerType	DMTF|IRQ|002.TriggerType	CIM_IRQ.TriggerType
IRQ.Shareable	DMTF|IRQ|002.Shareable	CIM_IRQ.Shareable
Keyboard.Layout	DMTF|Keyboard|003.Layout	CIM_Keyboard.Layout
Keyboard.Subtype	Not applicable	Not applicable
Keyboard.Type	DMTF|Keyboard|003.Keyboard.Type	CIM_Keyboard.Description
Keyboard.Fkeys	Not applicable	CIM_Keyboard.NumberOfFunctionKeys
Keyboard.Delay	Not applicable	Not applicable
Keyboard.TypematicRate	Not applicable	Not applicable
VideoAdapter.NumberOfColorPlanes (NEW)	Not applicable	Win32_VideoController.NumberOfColorPanes
VideoAdapter.HorizontalResolution	DMTF|Video|004.Current Horizontal Resolution	Win32_VideoController.CurrentHorizontalResolution
VideoAdapter.VerticalResolution	DMTF|Video|004.Current Vertical Resolution	Win32_VideoController.CurrentVerticalResolution
VideoAdapter.DisplayType	DMTF|Video|004.Video Type	Win32_VideoController.VideoArchitecture
VideoAdapter.MemoryType	DMTF|Video|004.Video Memory Type	Win32_VideoController.VideoMemoryType
VideoAdapter.MaxMemorySupported	DMTF|Video|004.Video RAM Memory Size	Win32_VideoController.AdapterRAM

Scan Data	DMI Class and Attribute	WMI Class and Attribute
VideoAdapter.Bitsperpixel	DMTF|Video|004.Current Number of Bits per Pixel	Win32_VideoController.CurrentBitsPerPixel
VideoAdapter.ControllerDescription	DMTF|Video|004.Video Controller Description	Win32_VideoController.Description
VideoAdapter.MaxRefreshrate	DMTF|Video|004.Maximum Refresh Rate	Win32_VideoController.MaxRefreshRate
VideoAdapter.MinRefreshrate	DMTF|Video|004.Minimum Refresh Rate	Win32_VideoController.MinRefreshRate
VideoAdapter.DACType	Not applicable	Win32_VideoController.AdapterDACType
VideoAdapter.ChipSet	Not applicable	Win32_VideoController.VideoProcessor
VideoAdapter.ProviderName	Not applicable	Win32_VideoController.VideoAdapterCompatibility
VideoBIOS.VideoBIOSManufacturer	DMTF|Video BIOS|001.BIOS Manufacturer	CIM_VideoBIOSElement.Manufacturer
VideoBIOS.VideoBIOSVersion	DMTF|Video BIOS|001.Video.BIOS Version	CIM_VideoBIOSElement.Version
VideoBIOS.VideoBIOSReleaseDate	DMTF|Video BIOS|001.Video.BIOS Release Date	CIM_VideoBIOSElement.InstallDate
VideoBIOS.VideoBIOS.IsShadowed	DMTF|Video BIOS|001.Video.Shadowing State	CIM_VideoBIOSElement.IsShadowed
ParallelPort.Name	DMTF|Parallel Ports|003.Parallel Port Index	CIM_ParallelController.Name
ParallelPort.DMASupport	DMTF|Parallel Ports|003.DMA Support	CIM_ParallelController.DMASupport
ParallelPort.Address	DMTF|Parallel Ports|003.Parallel Base I/O Address	Not applicable
ParallelPort.IRQ	DMTF|Parallel Ports|003.IRQ Used	Not applicable
SerialPort.Name	DMTF|Serial Ports|004.Serial Port Index	CIM_SerialController.Name
SerialPort.Address	DMTF|Serial Ports|004.Serial Base I/O Address	Not applicable
SerialPort.IRQ	DMTF|Serial Ports|004.IRQ Used	Not applicable

Scan Data	DMI Class and Attribute	WMI Class and Attribute
FloppyDrive.DeviceID	DMTF|Logical Drives|001.Logical Drive Name (when DMTF|Logical Drives|001.Logical Drive Type=Floppy Drive(7))	Win32_LogicalDisk.DeviceID (where Win32_LogicalDisk.DriveType = 2 (Removable Disk) and Win32_LogicalDisk.MediaType = [1,10])
FloppyDrive.Manufacture	Not applicable	Not applicable
FloppyDrive.Description	Hard Code: Floppy Drive (when DMTF|Disks|003.Storage Type=Floppy Disk(4))	Win32_LogicalDisk.Description (where Win32_LogicalDisk.DriveType = 2 (Removable Disk) and Win32_LogicalDisk.MediaType = [1,10])
FloppyDrive.MaxNumberofCylinders	Not applicable	Not applicable
FloppyDrive.NumberOfHeads	Not applicable	Not applicable
FloppyDrive.SectorsPerTrack	Not applicable	Not applicable
FloppyDrive.Size	DMTF|Logical Drives|001.Logical Drive Size (when DMTF|Logical Drives|001.Logical Drive Type = Floppy Drive(7))	Win32_LogicalDisk.Size (where Win32_LogicalDisk.DriveType = 2 (Removable Disk) and Win32_LogicalDisk.MediaType = [1,10])
CDROMDrive.DeviceID	DMTF|Logical Drives|001.Logical Drive Name (When DMTF|Logical Drives|001.Logical Drive Type = 6)	Win32_CDROMDrive.Drive
CDROMDrive.Manufacture	Not applicable	Win32_CDROMDrive.Manufacturer
CDROMDrive.Description	Not applicable	Win32_CDROMDrive.Description
CDROMDrive.Caption	Hard code: CDROM Device (when DMTF|Disks|001.Logical Drive Type = 6)	Win32_CDROMDrive.Caption
HardDrive.Media Type	DMTF|Disks|003.Removable Media	Win32_DiskDrive.MediaType
HardDrive.Vendor	Not applicable	Win32_DiskDrive.Manufacturer
HardDisk.Description	DMTF|Disks|003.Interface Description (when DMTF|Disks|003.Storage Type=Hard Disk(3))	Win32_DiskDrive.Description

Scan Data	DMI Class and Attribute	WMI Class and Attribute
HardDisk.Cylinders	DMTF|Disks|003.Number of Physical Cylinders	Win32_DiskDrive.TotalCylinders
HardDisk.Heads	DMTF|Disks|003.Number of Physical Heads	Win32_DiskDrive.TotalHeads
HardDisk.Sectors	DMTF|Disks|003.Number of Physical Sectors per Track	Win32_DiskDrive.SectorsPerTrack
HardDisk.Capacity	DMTF|Disks|003.Total Physical Size	Win32_DiskDrive.Size
LogicalDrive.Name	Not applicable	Win32_LogicalDiskDeviceID (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
LogicalDrive.VolumeSerialNumber	Not applicable	Win32_LogicalDisk.VolumeSerialNumber (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
LogicalDrive.Volume (Volume Label)	Not applicable	Win32_LogicalDisk.VolumeName (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
Operating System.OSType	Not applicable	Win32_OperatingSystem.OSType
OperatingSystem.Version	Not applicable	Not applicable
OperatingSystem.Codepage	Not applicable	Win32_OperatingSystem.CodeSet
OperatingSystem.InstallDate	Not applicable	Win32_OperatingSystem.InstallDate
OperatingSystem.TotalSwapSpaceSize	DMTF|System Memory Settings|001.Total Size of Paging Files	Win32_OperatingSystem.SizeStoredInPagingFiles
OperatingSystem.Description	DMTF|Operating System|001.Operating System Description	Win32_OperatingSystem.Caption
OperatingSystem.OtherTypeDescription	Not applicable	Win32_OperatingSystem.OtherTypeDescription
OperatingSystem.VirtualMemorySize	DMTF|System Memory Settings|001.Total Virtual Memory	Win32_OperatingSystem.TotalVirtualMemory
OperatingSystem.VisibleMemorySize	Not applicable	Win32_OperatingSystem.TotalVisibleMemorySize
OperatingSystem.Role	Not applicable	Not applicable
InventoryScanner.Version	Not applicable	Not applicable
InventoryScanner.LastScanDate	Not applicable	Not applicable
InventoryScanner.InventoryServer	Not applicable	Not applicable

Scan Data	DMI Class and Attribute	WMI Class and Attribute
InventoryScanner.ScanMode	Not applicable	Not applicable
InventoryScanner.GeneralDictionaryVersion	Not applicable	Not applicable
InventoryScanner.PrivateDictionaryVersion	Not applicable	Not applicable
SoundCard.Description	Not applicable	Win32_SoundDevice.Description
SoundCard.Name	Not applicable	Win32_SoundDevice.Name
SoundCard.Manufacturer	Not applicable	Win32_SoundDevice.Manufacturer
Cache.Level	DMTF|System Cache|003.System Cache Level	Win32_CacheMemory.Level
Cache.WritePolicy	DMTF|System Cache|003.System Cache Write Policy	Win32_CacheMemory.WritePolicy
Cache.ErrorCorrection	DMTF|System Cache|003.System Cache Error Correction	Win32_CacheMemory.ErrorCorrectType
Cache.Type	DMTF|System Cache|003.System Cache Type	Win32_CacheMemory.CacheType
Cache.LineSize	DMTF|System Cache|003.Line Size	Win32_CacheMemory.LineSize
Cache.ReplacementPolicy	DMTF|System Cache|003.Replacement Policy	Win32_CacheMemory.ReplacementPolicy
Cache.ReadPolicy	DMTF|System Cache|003.Read Policy	Win32_CacheMemory.ReadPolicy
Cache.Associativity	DMTF|System Cache|003.Associativity	Win32_CacheMemory.Associativity
Cache.Speed	DMTF|System Cache|003.System Cache Speed	Win32_CacheMemory.CacheSpeed
Cache.Size	DMTF|System Cache|003.System Cache Size	Win32_CacheMemory.MaxCacheSize
MotherBoard.Version	Not applicable	Win32_BaseBoard.Version
MotherBoard.Description	Not applicable	Win32_BaseBoard.Description
MotherBoard.Slots	DMTF|Motherboard|001.Number of Expansion slots	Not applicable
MotherBoard.Manufacture	Not applicable	Win32_BaseBoard.Manufacture

Scan Data	DMI Class and Attribute	WMI Class and Attribute
Battery.Name	DMTF|Portable Battery|002.Portable Battery Device Name	Win32_Battery.Name
Battery.Chemistry	DMTF|Portable Battery|002.Portable Battery Device Chemistry	Win32_Battery.Chemistry
Battery.Capacity	DMTF|Portable Battery|002.Portable Battery Design Capacity	Win32_Battery.DesignCapacity
Battery.Voltage	DMTF|Portable Battery|002.Portable Battery Design Voltage	Win32_Battery.DesignVoltage
Battery.Version	DMTF|Portable Battery|002.Portable Battery Smart Battery Version	Win32_Battery.SmartBatteryVersion
Battery.Manufacturer	DMTF|Portable Battery|002.Portable Battery Manufacturer	Win32_PortableBattery.Manufacturer
Battery.ManufactureDate	DMTF|Portable Battery|002.Portable Battery Manufacturer Date	Win32_Battery.InstallDate
Battery.SerialNumber	DMTF|Portable Battery|002.Portable Battery Serial Number	Not applicable
PowerSupply.InputVoltageDescription	DMTF|Power Supply|002.Power Supply Input Voltage Capability Description	CIM_UninterruptiblePowerSupply.Description
PowerSupply.Power	DMTF|Power Supply|002.Total Output Power	CIM_UninterruptiblePowerSupply.TotalOutputPower
DMA.Number	DMTF|DMA|001.DMA Number	CIM_DMA.DMAChannel
DMA.Description	DMTF|DMA|001.DMA Description	CIM_DMA.Description
DMA.Availability	DMTF|DMA|001.DMA Channel Availability	CIM_DMA.Availability
DMA_BurstMode	DMTF|DMA|001.DMA BurstMode	CIM_DMA.BurstMode
UCS.DNName	Not applicable	Not applicable

Scan Data	DMI Class and Attribute	WMI Class and Attribute
UCS.PrimaryOwnerContact	DMTF|General Information|001.3	CIM_UnitaryComputerSystem.PrimaryOwnerContact
UCS.PrimaryOwnerName	DMTF|General Information|001.4	CIM_UnitaryComputerSystem.PrimaryOwnerName
PointingDevice.DeviceType	DMTF|Pointing Device|Pointing Device Type(1)	CIM_PointingDevice.PointingType
PointingDevice.Type	DMTF|Pointing Device|Pointing Device Interface (2)	CIM_PointingDevice.Name
PointingDevice.NumberOfButtons	DMTF|Pointing Device|Pointing Device Buttons (4)	CIM_PointingDevice.NumberOfButtons
PointingDevice.DriverName	DMTF|Pointing Device|Pointing Device Driver Name (6)	Not applicable
PointingDevice.DriverVersion	DMTF|Pointing Device|Pointing Device Driver Version (7)	CIM_PointingDevice.Name
PointingDevice.IRQ	DMTF|Pointing Device|Pointing Device IRQ (3)	Not applicable
Slot.Description	DMTF|System Slots|003.Description	Win32_SystemSlot. SlotDesignation
Slot.MaxDataWidth	DMTF|System Slots|003.MaxDataWidth	Win32_SystemSlot. MaxDataWidth
Slot.ThermalRating	DMTF|System Slots|003.Slot Thermal Rating	Win32_SystemSlot. ThermalRating
FileSystem.Drive	Not applicable	Win32_LogicalDiskDeviceID (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
FileSystem.FileSystemSize	Not applicable	Win32_LogicalDisk.Size (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
FileSystem.AvailableSpace	Not applicable	Wind32_LogicalDisk.FreeSpace (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
FileSystem.FileSystem	Not applicable	Win32_LogicalDisk.FileSystem (when Win32_LogicalDisk.DriveType = 3 (Local Disk))

Scan Data	DMI Class and Attribute	WMI Class and Attribute
Monitor.Device ID	Not applicable	Not applicable
Monitor.Description	Not applicable	Not applicable
Monitor.Manufacturer Date	Not applicable	Not applicable
Monitor.Model ID	Not applicable	Not applicable
Monitor.ViewableSize (inches)	Not applicable	Not applicable
Monitor.NominlaSize (inches)	Not applicable	Not applicable
Monitor.Serial Number	Not applicable	Not applicable
Monitor.Manufacturer	Not applicable	Not applicable
Monitor.Model	Not applicable	Not applicable
Chassis.Type (enum)	DMTF|Physical Container Global Table|1	Win32_SystemEnclosure. ChassisTypes
Chassis.Manufacturer	DMTF|FRU||4	Win32_SystemEnclosure. Manufacturer
Chassis.SerialNumber	DMTF|FRU||7	Win32_SystemEnclosure. SerialNumber
Chassis.AssetTag	DMTF|Physical Container Global Table|2	Win32_SystemEnclosure. SMBIOSAssetTag
Chassis.Version	Not applicable	Win32_SystemEnclosure. Version
Chassis.NumberOfPowerCords	Not applicable	Win32_SystemEnclosure. NumberOfPowerCords
Chassis.Tag	Not applicable	Win32_SystemEnclosure.Tag

NOTE: PCMCIA modems are connected to the computer through the PCMCIA slots on the inventoried servers. The Scanner detects PCMCIA modems that are active on the computer. If you want to know which modem is installed on the computer, use the Windows System Device Manager on the Windows server.

Non-PCMCIA modems are connected to the computer through the external ports. For example, some non-PCMCIA modems are connected through the serial ports. The Scanner detects non-PCMCIA modems that are installed on the computer.

Non-PCMCIA modems might not be active at the time of scanning. Also, these modems might not be connected, although they are configured on the computer. In this case, the Scanner detects the modem and reports the scan information of the modem.

The Inventory scanner reports inventory information for the monitors that are manufactured only after 1997.

J ZENworks 6.5 Server Management Inventory Attributes

The following table lists the Server Inventory attributes that ZENworks 6.5 Server Management uses.

Each row in the table has:

- Name of the attribute as displayed in the Inventory Database Export Wizard in ConsoleOne
- Name of the attribute in the exported .csv file (first row in the .csv file)
- Inventory database attribute name
- Type of the attribute in the Inventory database
- Length of the attribute in the Inventory database
- Brief description of the attribute

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
General-NDSName-Label	NDSName_LABEL	ManageWise.NDSName.Label	String	254	The DN name of the inventoried server registered in eDirectory.
SystemInfo.Description	Asset_Description	Zenworks.SystemInfo.Description	String	254	Description of the system asset information.
SystemInfo.Caption	Asset_Caption	Zenworks.SystemInfo.Caption	String	64	Identifying information of the computer.
SystemInfo.Tag	Asset_Asset Tag	Zenworks.SystemInfo.Tag	String	254	Asset tag number that the ROM-based setup program creates. This is unique to every inventoried server.
SystemInfo.ModelNumber	Asset_Model Number	Zenworks.SystemInfo.Model	String	64	Model number value for the computer, assigned during manufacture.
SystemInfo.SerialNumber	Asset_Serial Number	Zenworks.SystemInfo.SerialNumber	String	64	Model serial number value for the computer, assigned during manufacture.
SystemInfo.ManagementTechnology	Asset_Management Technology	Zenworks.SystemInfo.ManagementTechnology	Integer		The management technology available on the computer system.
CurrentLoginUser.Name	Current Login User.Name	ManageWise."User".Name	String	254	User logged in to the Primary eDirectory tree when the inventoried server was scanned.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
LastLoginUser.Name	Last Login User.Name	ManageWise."User".Name	String	254	User most recently logged into the Primary eDirectory tree through Novell Client when the inventoried server was scanned.
Product.Name	Applications_Name	CIM.Product.Name	String	254	Name of the software application.
Product.Vendor	Applications_Vendor	CIM.Product.Vendor	String	254	Name of the software application manufacturer.
Product.Version	Applications_Version	CIM.Product.Version	String	64	Version of the software application.
Product.Location	Applications_Path	CIM.Directory.Location	String	254	The product installation path.
Product.IdentifyingNumber	Applications_Identifying Number	CIM.Product.IdentifyingNumber	String	64	Microsoft product ID
WinOperatingSystem.OSType	Windows_Name	ZENworks.WINOperatingSystem.OSType	Unsigned Small Integer (enum)		Operating system name. For example, Windows 2000. See “Enumeration Values for Software-Operating Systems-Windows - Name” on page 724.
WinOperatingSystem.Version	Windows_Version	ZENworks.WINOperatingSystem.Version	String	254	Version of the operating system.
WinOperatingSystem.Caption	Windows_Caption	ZENworks.WINOperatingSystem.Caption	String	64	Short name of the operating system. For example, Windows 2000.
WinOperatingSystem.Role	Windows_Role	ZENworks.WINOperatingSystem.Role	Integer (enum)		The role of the computer system. For example, server.
WinOperatingSystem.OtherTypeDescription	Windows_Other Description	ZENworks.WINOperatingSystem.Description	String	254	More description about the operating system.
WinOperatingSystem.InstallDate	Windows_Install Date	ZENworks.ZENOperatingSystem.InstallDate	String	25	Installation date of the operating system.
WinOperatingSystem.CodePage	Windows_Code Page	ZENworks.WINOperatingSystem.CodePage	String	254	Current language code page being used.
WinOperatingSystem.TotalVisibleMemorySize	Windows_Total Memory (MB)	ZENworks.WINOperatingSystem.TotalVisibleMemorySize	Integer		Total memory as reported by the Windows operating system.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
WinOperating System.TotalVirtual MemorySize	Windows_Total Virtual Memory (MB)	ZENworks.WINOperatingSystem.TotalVirtualMemorySize			Total virtual memory as reported by the Windows operating system.
InventoryScanner. Version	Scanner Information_Version	ZENworks.InventoryScanner.Version	String	64	Version of the Scanner running on the inventoried server.
InventoryScanner. LastScanDate	Scanner Information_Last Scan Date	ZENworks.InventoryScanner.LastScanDate	Unsigned Integer		The date when the Scanner was last scanned. Stored as milliseconds time value so it can be read and displayed in any appropriate date format.
InventoryScanner.Inventory Server	Scanner Information_Inventory Server	ZENworks.InventoryScanner.InventoryServer	String	254	Name of the Inventory server to which the scans are sent. It is not the complete DN of the server name.
InventoryScanner. ScanMode	Scanner Information_Scan Mode	ZENworks.InventoryScanner.ScanMode	Integer (enum)		The management technology used by the Scanner, such as WMI or DMI, for scanning the computer system.
NetWareClient.Version	Netware Client_Version	ZENworks.NetWareClient.Version	String	64	Version of the NetWare client software installed on the inventoried server.
NetworkAdapterDriver.Description	Network Adapter Driver_Description	ZENworks.NetworkAdapterDriver.Description	String	254	Description of the network adapter driver installed on the inventoried server. For example, IBM 10/100 Ethernet adapter, EN-2420Px Ethernet adapter.
NetworkAdapterDriver.Name	Network Adapter Driver_Name	ZENworks.NetworkAdapterDriver.Name	String	254	Name of the network adapter driver software installed that corresponds to the adapter. For example, ne2000.sys, pppmac.vxd, and others.
NetworkAdapterDriver.Version	Network Adapter Driver_Version	ZENworks.NetworkAdapterDriver.Version	String	64	Network adapter driver version.
PointingDeviceDeviceDriver.Name	Pointing Device Driver_Name	ZENworks.PointingDeviceDeviceDriver.Name	String	254	Name of the mouse driver installed on the inventoried server.
PointingDeviceDeviceDriver.Version	Pointing Device Driver_Version	ZENworks.PointingDeviceDeviceDriver.Version	String	64	Mouse driver version.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
PointingDevice.Name	Pointing Device_Name	CIM.PointingDevice.Name	String	254	The name of the pointing device, such as Mouse. The string stored in this field will be MOUSE. The CIM.PointingDevice.PointingType field determines the type of the pointing device. The different types of pointing devices are as listed in “Enumeration Values for Hardware-Pointing Device-Name” on page 725.
PointingDevice.Numberofbuttons	Pointing Device_Number of Buttons	CIM.PointingDevice.NumberOfButtons	Unsigned Tiny Integer		The number of buttons used by the pointing device.
PointingDevice.IRQNumber	Pointing Device_IRQ Number	CIM.IRQ.IRQNumber	Unsigned Integer		The IRQ channel on the system to which the Mouse pointing device is attached. This information is stored in an IRQ class and not in the PointingDevice class in the database. For more information on how they are associated, see “Understanding the ZENworks 6.5 Server Managements Inventory Database Schema” on page 513.
PointingDevice.PointingType	Pointing Device_Type	CIM.PointingDevice.PointingType	Integer (enum)		The pointing device type.
ZENKeyboard.Numberoffunctionkeys	Keyboard_Number of Function Keys	ZENworks.ZENKeyboard.NumberOfFunctionKeys	Unsigned Small Integer		Number of function keys on keyboard.
ZENKeyboard.Layout	Keyboard_Layout	ZENworks.ZENKeyboard.layout	String	254	Layout information. For example, US English.
ZENKeyboard.SubType	Keyboard_Subtype	ZENworks.ZENKeyboard.SubType	Unsigned Integer		A number indicating the subtype of the keyboard.
ZENKeyboard.Delay	Keyboard_Delay (mSecs)	ZENworks.ZENKeyboard.Delay	Unsigned Integer		Delay before the repeat of a key.
ZENKeyboard.Typematicrate	Keyboard_Typematic Rate (mSecs)	ZENworks.ZENKeyboard.Typematic Rate	Unsigned Integer		Rate of processing the keys.
ZENKeyboard.Description	Keyboard_ Description	ZENworks.ZENKeyboard.Description	String	254	Keyboard description indicating the type of keyboard. For example, IBM enhanced (101/102 key) keyboard.
VideoBIOSElement.Manufacturer	Display Driver_ Manufacturer	CIM.VideoBIOSElement.Manufacturer	String	254	Manufacturer of the video BIOS driver installed on the system.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
VideoBIOSElement.Version	Display Driver_Version	CIM.VideoBIOSElement.Version	String	254	Version of the Video BIOS driver.
VideoBIOSElement.Install Date	Display Driver_Install Date	CIM.VideoBIOSElement.InstallDate	String	25	Video BIOS release date.
VideoBIOSElement.IsShadowed	Display Driver_Is Shadowed	CIM.VideoBIOSElement.ISShadowed	BIT (Used for Boolean conditioners)		A Boolean condition indicating if the video BIOS supports shadow memory. 0 represents False and 1 is True.
VideoAdapter.NumberOfcolorpanes	Display Adapter_ Number of Color Planes	ZENworks. VideoAdapter.NumberOf ColorPlanes	Unsigned Integer		Number of color planes supported by the video system.
VideoAdapter.CurrentVerticalResolution	Display Adapter_ Current Vertical Resolution	ZENworks. VideoAdapter.Current Vertical Resolution	Unsigned Integer		Vertical resolution of the display.
VideoAdapter.CurrentHorizontalResolution	Display Adapter_ Current Horizontal Resolution	ZENworks. VideoAdapter.Current Horizontal Resolution	Unsigned Integer		Horizontal resolution of the display.
VideoAdapter. Description	Display Adapter_ Description	ZENworks. VideoAdapter.Description	String	254	Video adapter description.
VideoAdapter.MinRefreshRate	Display Adapter_ Minimum Refresh Rate	ZENworks. VideoAdapter.Min Refresh Rate	Unsigned Integer		Minimum refresh rate of the monitor for redrawing the display, measured in Hertz.
VideoAdapter.MaxRefreshRate	Display Adapter_ Maximum Refresh Rate	ZENworks. VideoAdapter.Max Refresh Rate	Unsigned Integer		Maximum refresh rate of the monitor for redrawing the display, measured in Hertz.
VideoAdapter.VideoArchitecture	Display Adapter_ Video Architecture	ZENworks. VideoAdapter.Video Architecture	Unsigned Integer (enum)		The architecture of the video subsystem in this system. For example, CGA/VGA/SVGA/8514A. See “Enumeration Values for Hardware-Display Adapter-Video Architecture” on page 725.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
VideoAdapter.VideoMemoryType	Display Adapter_ Video Memory Type	ZENworks. VideoAdapter.VideoMemoryType	Unsigned Small Integer (Enum)		The type of memory for this adapter. For example, VRAM/SRAM/DRAM/ EDO RAM. See “Enumeration Values for Hardware-Display Adapter-Video Memory Type” on page 725.
VideoAdapter.Maxmemorysupported	Display Adapter_ Maximum Memory Supported(KB)	ZENworks. VideoAdapter.Max Memory Supported	Unsigned Integer		Maximum memory that the display adapter supports for VIDEO RAM.
VideoAdapter.CurrentBitsPerPixel	Display Adapter_ Current Bits/Pixel	ZENworks. VideoAdapter.Current Bits PerPixel	Unsigned Integer		Number of adjacent color bits for each pixel.
VideoAdapter.Chip Set	Display Adapter_ Chip Set	ZENworks. VideoAdapter.Chip Set	String	254	The chip set used in the video adapter.
VideoAdapter.DAC Type	Display Adapter_ DAC Type	ZENworks. VideoAdapter.DAC Type	String	254	The digital to analog converter type used in the video adapter.
VideoAdapter.ProviderName	Display Adapter_ Provider	ZENworks.VideoAdapter.Provider	String	254	The manufacturer or the provider name.
ZENPOTSModem. Caption	Modem_Caption	ZENworks.ZENPOTSModem. Caption	String	64	The short name of the modem.
ZENPOTSModem. Description	Modem_Description	ZENworks.ZENPOTSModem. Description	String	254	The complete description of the modem. For example, Standard 2400 bps modem, IBM PCMCIA HPC modem.
ZENPOTSModem. Name	Modem_Name	ZENworks.ZENPOTSModem.Name	String	254	The name of the modem dictating its type and usage. For example, Standard Windows Modem means that this is used in standard Windows architecture.
ZENPOTSModem. ProviderName	Modem_Provider	ZENworks.ZENPOTSModem.Provider	String	254	The manufacturer or the provider name.
ZENPOTSModem. DeviceID	Modem_Device ID	ZENworks.ZENPOTSModem.DeviceID	String	64	The unique ID assigned to the device.
BIOS.BIOSIDBytes	BIOS_BIOS Identification Bytes	ZENworks. BIOS.BIOS IDBytes	String	254	Byte in the BIOS that indicates the computer model.
BIOS.SerialNumber	BIOS_ Serial Number	ZENworks. BIOS.Serial Number	String	64	Serial number of BIOS assigned by the manufacturer.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
BIOS.PrimaryBIOS	BIOS_Primary Bios	ZENworks. BIOS.PrimaryBIOS	BIT (Used for Boolean conditions here)		True when set to 1, indicating that this BIOS is the primary BIOS. Used in systems with additional BIOS chips.
BIOS.InstallDate	BIOS_Install Date	ZENworks. BIOS.Install Date	String	25	The release date of the BIOS given by the manufacturer.
BIOS.Version	BIOS_Version	ZENworks. BIOS.Version	String	254	Version or revision level of the BIOS.
BIOS. Manufacturer	BIOS_ Manufacturer	ZENworks. BIOS. Manufacturer	String	254	The manufacturer name of BIOS.
BIOS.Caption	BIOS_Caption	ZENworks. BIOS.Caption	String	64	The name of the BIOS as given by the BIOS manufacturer.
BIOS."size"	BIOS_Size(KB)	ZENworks. BIOS.size	Unsigned Integer		Size of the BIOS in bytes.
Processor.Current ClockSpeed	Processor_Current Clock Speed(MHz)	CIM. Processor. CurrentClockSpeed	Unsigned Integer		Current clock speed of the processor in MHz.
Processor.Maxclockspeed	Processor_ Maximum Clock Speed(MHz)	CIM. Processor. MaxClock Speed	Unsigned Integer		Maximum clock speed of the processor in MHz.
Processor.Role	Processor_Role	CIM. Processor. Role	String	254	Type of processor such as central processor, math coprocessor, and others
Processor.Family	Processor_ Processor Family	CIM. Processor. Family	Unsigned Small Integer (enum)		Family the processor belongs to. See “Enumeration Values for Hardware-Processor-Processor Family” on page 726.
Processor.Otherfamilydescription	Processor_Other Family Description	CIM. Processor. OtherFamily Description	String	64	Additional description about the processor family, such as the Pentium processor with MMX technology when the processor cannot be designated using Family.
Processor.Upgrade Method	Processor_ Upgrade Method	CIM. Processor. Upgrade Method	Unsigned Small Integer (Enum)		The method by which this processor can be upgraded, if upgrades are supported. See “Enumeration Values for Hardware-Processor-Upgrade Method” on page 726.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
Processor.Stepping	Processor_ Processor Stepping	CIM. Processor. Stepping	String	254	Single-byte code characteristic provided by microprocessor vendors to identify the processor stepping model.
Processor.Device ID	Processor_ DeviceID	CIM. Processor. DeviceID	String	64	Special hexadecimal string identifying the processor type.
CacheMemory.Speed	Cache Memory_ Speed(nsec)	CIM.PhysicalMemory. Speed	Unsigned Integer		Speed of this System Cache module in nanoseconds. This is stored in CIM.PhysicalMemory class and is associated to CIM.CacheMemory. For more information on how they are associated, see “Understanding the ZENworks 6.5 Server Managements Inventory Database Schema” on page 513.
CacheMemory.Capacity	Cache Memory_ Capacity(MB)	CIM.PhysicalMemory. Capacity	Unsigned Integer		Capacity of this System Cache module in nanoseconds. This is stored in CIM.PhysicalMemory class and is associated to CIM.CacheMemory. For more information on how they are associated, see “Understanding the ZENworks 6.5 Server Managements Inventory Database Schema” on page 513.
CacheMemory.Level	Cache Memory_ Level	CIM.Cache Memory. "Level"	Unsigned Small Integer (enum)		Indicates the cache level: internal cache that is built in to the microprocessors, or external cache that is between the CPU and DRAM.
CacheMemory.WritePolicy	Cache Memory_ Write Policy	CIM.Cache Memory. WritePolicy	Unsigned Small Integer (enum)		Indicates the two different ways (Write-Back and Write-Through Cache) that the cache can handle to write to the memory.
CacheMemory.Errormethodology	Cache Memory_ Error Methodology	CIM.CacheMemory.Error Methodology	String	254	Error correction scheme supported by this cache component, for example, Parity/Single Bit ECC/MultiBit ECC.
CacheMemory.Cachetype	Cache Memory_ Cache Type	CIM.Cache Type	Unsigned Small Integer (enum)		Defines the system cache type. For example, Instruction, Data, Unified.
CacheMemory.LineSize	Cache Memory_ Line Size (Bytes)	CIM.Cache Memory .LineSize	Unsigned Integer		Size in bytes of a single cache bucket or line.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
CacheMemory.ReplacementPolicy	Cache Memory_ Replacement Policy	CIM.Cache Memory. ReplacementPolicy	Unsigned Integer (enum)		Algorithm that the cache uses to determine which cache lines or buckets should be reused.
CacheMemory.ReadPolicy	Cache Memory_ Read Policy	CIM.Cache Memory. ReadPolicy	Unsigned Small Integer (enum)		Indicates whether the data cache is for read operation.
CacheMemory.Associativity	Cache Memory_ Associativity	CIM.Cache Memory. Associativity	Unsigned Integer (enum)		Defines the system cache associativity (direct-mapped, 2-way, 4-way).
Diskette Drive.Manufacturer	Diskette Drive_ Manufacturer	ZENworks. Physical Diskette. Manufacturer	String	254	Vendor name.
Diskette Drive.Description	Diskette Drive_ Description	ZENworks. Physical Diskette. Description	String	254	Floppy diskette description.
Diskette Drive.PhysicalCylinders	Diskette Drive_Physical Cylinders	ZENworks. Physical Diskette. Physical Cylinders	Unsigned Integer		Total number of cylinders or tracks on the floppy.
Diskette Drive.PhysicalHeads	Diskette Drive_Physical Heads	ZENworks. Physical Diskette. Physical Heads	Unsigned Small Integer		Number of heads.
Diskette Drive.Capacity	Diskette Drive_Capacity (MB)	ZENworks. Physical Diskette. Capacity	Unsigned Integer		Total size.
Diskette Drive.SectorsPerTrack	Diskette Drive_Sectors/Track	ZENworks. Physical Diskette. SectorsPer Track	Unsigned Integer		Number of sectors per track.
Diskette Drive. DeviceID	Diskette Drive_ DeviceID	CIM.Diskette Drive	String	64	The drive name representing the floppy drive.
ZENDiskDrive.Manufacturer	Physical Disk Drive_ Manufacturer	ZENworks. PhysicalDisk.Manufacturer	String	254	Vendor name.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
ZENDiskDrive.Description	Physical Disk Drive_ Description	ZENworks. PhysicalDisk.Description	String	254	Hard disk vendor description.
ZENDiskDrive.PhysicalCylinders	Physical Disk Drive_ Physical Cylinders	ZENworks. PhysicalDisk.Physical Cylinders	Unsigned Integer		Total number of cylinders.
ZENDiskDrive.PhysicalHeads	Physical Disk Drive_Physical Heads	ZENworks. PhysicalDisk.Physical Heads	Unsigned Small Integer		Number of heads.
ZENDiskDrive.SectorsPerTrack	Physical Disk Drive_Sectors/Track	ZENworks. PhysicalDisk.SectorsPer Track	Unsigned Integer		Number of sectors per track.
ZENDiskDrive.Capacity	Physical Disk Drive_ Capacity(MB)	ZENworks. PhysicalDisk.Capacity	Unsigned Integer		Total size of the hard disk.
ZENDiskDrive.Removable	Physical Disk Drive_ Removable	ZENworks.LogicalDiskDrive.Removable	BIT		0 indicates that it is a fixed disk and 1 indicates that it is a removable disk.
LocalFileSystem.DeviceID	Logical Disk Drive_ Device ID	ZENworks.LogicalDiskDrive.DeviceID	String	64	The drive letter, such as C: or A:.
LocalFileSystem.FileSystemSize	Logical Disk Drive_ Size(MB)	CIM.LocalFileSystem.FileSystemSize	Integer		The total size of the file system or the logical disk.
LocalFileSystem.AvailableSpace	Logical Disk Drive_ Free Size(MB)	CIM.LocalFileSystem.AvailableSpace	Integer		The available size of the file system or the logical disk.
LocalFileSystem.VolumeSerialNumber	Logical Disk Drive_ Volume Serial Number	CIM.LocalFileSystem.VolumeSerialNumber	String	254	The volume serial number of the specified drive.
LocalFileSystem.Caption	Logical Disk Drive_ Caption	CIM.LocalFileSystem.Caption	String	64	The volume label of the specified drive.
LocalFileSystem.FileSystemType	Logical Disk Drive_ File System Type	CIM.LocalFileSystem.FileSystemType	String	254	The file system on the drive, such as FAT or NTFS.
CDROMDrive.Manufacturer	CDROM_Manufacturer	ZENworks. Physical CDROM. Manufacturer	String	254	The manufacturer of the CD-ROM drive.
CDROMDrive.Caption	CDROM_Caption	ZENworks. Physical CDROM. Caption	String	64	CD-ROM label.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
CDROMDrive.Description	CDROM_ Description	ZENworks. Physical CDROM. Description	String	254	Description of the CD drive, as given by the manufacturer. For example, ATAPI CDROM, CREATIVE CD1620E SL970520.
CDROMDrive.DeviceID	CDROM_ Device ID	ZENworks. Logical CDROM. DeviceID	String	64	Drive letter allocated for the CD on the inventoried server.
SerialPort.Name	Serial Port_Name	ZENworks. SerialPort. Name	String	254	The name of the serial port. For example, COM1, COM2, and others.
SerialPort.Address	Serial Port_ Address	ZENworks. SerialPort. Address	Unsigned Integer		The address mapped in memory for the serial port.
SerialPort.IRQNumber	Serial Port_IRQ Number	CIM.IRQ.IRQNumber	Unsigned Integer		The IRQ channel on the system to which the serial port is attached. In the database, this information is stored in an IRQ class and not in a Serial Port class. For more information on how they are associated, see Chapter 15, “Understanding the ZENworks 6.5 Server Managements Inventory Database Schema,” on page 513.
ParallelPort.Name	Parallel Port_Name	ZENworks. ParallelPort. Name	String	254	The name of the parallel port. For example, LPT1 and others.
ParallelPort.Address	Parallel Port_ Address	ZENworks. ParallelPort. Address	Unsigned Integer		The name of the parallel port. For example, LPT1 and others.
ParallelPort.DMASupport	Parallel Port_DMA Support	ZENworks. ParallelPort. DMASupport	BIT (used for Boolean conditions here)		If True or 1, then it means that DMA is the channel that is allocated for bulk data transfer for use with devices connected to the parallel ports.
ParallelPort.IRQNumber	Parallel Port_IRQ Number	CIM.IRQ. IRQNumber	Unsigned Integer		The IRQ channel on the system to which the parallel port is attached. This information is stored in an IRQ class and not in a parallel port class in the database. For more information on how they are associated, see Chapter 15, “Understanding the ZENworks 6.5 Server Managements Inventory Database Schema,” on page 513.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
Bus.Version	Bus_Version	ZENworks. Bus.Bus Version	String	254	Version of the bus supported by the inventoried server.
Bus.Description	Bus_Description	ZENworks.Bus.De scription	String	254	Description of the bus.
Bus.BusType	Bus_Bus Type	ZENworks.Bus.Bu sType	Integer (enum)		The bus type of the system.
Bus.Name	Bus_Name	ZENworks.Bus.Na me	String	254	Name of the internal system bus.
Bus.DeviceID	Bus_Device ID	ZENworks.Bus.De viceID	String	64	The unique ID for the specific bus.
ZENNetworkAdapt er.Name	Network Adapter_ Name	CIM.ZENworks.ZE NAdapter.Name	String	254	Network adapters installed on the system.
ZENNetworkAdapt er.MaxSpeed	Network Adapter_Max_Spee d (Mbps)	CIM.ZENworks.ZE NAdapter. MaxSpeed	Unsigned Integer		Rate at which the adapter can transfer data.
ZENNetworkAdapt er.PermanentAddr ess	Network Adapter_ Permanent Address	CIM.ZENworks.ZE NAdapter. PermanentAddress	String	64	Machine address stored permanently in the adapter (MAC address).
ZENNetworkAdapt er.MACAddress	Network Adapter_ Address	CIM.ZENworks.ZE NAdapter. MACAddress	String	64	The MAC address stored in the network adapter.
ZENNetworkAdapt er.ProviderName	Network Adapter_ Provider	CIM.ZENworks.ZE NAdapter. Provider	String	254	The manufacturer or the provider.
ZENNetworkAdapt er.AdapterType	Network Adapter_ Adapter Type	CIM.ZENworks.ZE NAdapter. AdapterType	String	254	Type of the adapter, such as Ethernet or FDDI adapter.
SoundAdapter.Des cription	Multimedia Card_ Description	ZENworks. SoundAdapter. Description	String	254	Description of the multimedia component for the inventoried server.
SoundAdapter.Na me	Multimedia Card_ Name	ZENworks. SoundAdapter. Name	String	254	Name of the sound card installed on the system.
SoundAdapter.Man ufacturer	Multimedia Card_ Manufacturer	ZENworks. SoundAdapter. Manufacturer	String	254	Vendor name.
SoundAdapter.Pro viderName	Multimedia Card_ Provider	ZENworks. SoundAdapter. Provider	String	254	The provider or the manufacturer of the multimedia card.
Battery.Name	Battery_Name	CIM.Battery. Name	String	254	Name of the battery installed on the system.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
Battery.Chemistry	Battery_Chemistry	CIM.Battery. Chemistry	Unsigned Small Integer		Indicates the battery's chemistry, such as lead acid, nickel cadmium and others. See "Enumeration Values for Hardware-Battery-Chemistry" on page 725.
Battery.DesignCapacity	Battery_Design Capacity(mWatt-hours)	CIM.Battery. Design Capacity	Unsigned Integer		The design capacity of the battery in mWatt-hours.
Battery.DesignVoltage	Battery_Design Voltage(MilliVolts)	CIM.Battery. DesignVoltage	Unsigned Integer		The design voltage of the battery in mVolts.
Battery.SmartBatteryVersion	Battery_ Smart Battery Version	CIM.Battery. SmartBatteryVersion	String	64	The Smart Battery Data Specification version number supported by this battery.
Battery.Manufacturer	Battery_ Manufacturer	CIM.PhysicalComponent. Manufacturer	String	254	Vendor name of the battery.
Battery.InstallDate	Battery_Install Date	CIM.PhysicalComponent. InstallDate	String	25	Date of manufacturing the battery.
Battery.SerialNumber	Battery_Serial Number	CIM.PhysicalComponent. SerialNumber	String	64	Battery serial number.
PowerSupply.Description	Power Supply_ Description	CIM.Power Supply. Description	String	254	Name and description of the power supply on the system.
PowerSupply.Total OutputPower	Power Supply_Total Output Power (MilliWatts)	CIM.Power Supply.Total OutputPower	Unsigned Integer		Total output power of the power supply.
IPProtocolEndPoint. Address	IP Address_ Address	CIM.IP Protocol Endpoint. Address	String	254	IP address of the inventoried server.
IPProtocolEndPoint.Subnet Mask	IP Address_ Subnet Mask	CIM.IP Protocol Endpoint. SubnetMask	String	254	The subnet mask of the inventoried server.
DNSName.LABEL	DNS_LABEL	ManageWise.DNS Name. Label	String	254	DNS name of the inventoried server.
IPXProtocolEndPoint.Address	IPX Address_ Address	CIM.IPX Protocol Endpoint. Address	String	254	IPX address of the inventoried server.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
LANEndPoint. MACAddress	MAC Address_ Address	CIM.LAN Endpoint. MACAddress	String	12	MAC address of the inventoried server.
MotherBoard.Version	MotherBoard_ Version	ZENworks.Motherboard.Version	String	64	Motherboard version.
MotherBoard.Description	MotherBoard_ Description	ZENworks.Motherboard.Description	String	254	The description of the motherboard.
MotherBoard.Manufacturer	MotherBoard_ Manufacturer	ZENworks.Motherboard.Manufacturer	String	254	The manufacturer of the motherboard.
MotherBoard.NumberOfSlots	MotherBoard_ Number Of Slots	ZENworks.Motherboard.Numberofslots	Integer		The number of expansion slots on the motherboard.
IRQ.Number	IRQ_IRQ Number	CIM.IRQ.IRQNumber	Unsigned Integer		The system interrupt number.
IRQ.Availability	IRQ_Availability	CIM.IRQ. Availability	Unsigned Small Integer (Enum)		Indicates whether the IRQ channel is used or available. Enumeration values are as follows: 1 = "Other" 2 = "Unknown" 3 = "Available" 4 = "In Use/Not Available" 5 = "In Use and Available/Shareable"
IRQ.TriggerType	IRQ_IRQ Trigger Type	CIM.IRQ. TriggerType	Unsigned Small Integer		IRQ trigger type indicating whether edge (value=4) or level triggered (value=3) interrupts occur. Enumeration values are as follows: 1 = "Other" 2 = "Unknown" 3 = "Level" 4 = "Edge"
IRQ.Shareable	IRQ_IRQ Shareable	CIM.IRQ. Shareable	Unsigned Small Integer		Boolean indicating whether the IRQ can be shared.
SLOT.MaxDataWidth	Slot_Maximum Data Width	CIM.Slot. MaxData Width	Unsigned Small Integer		Maximum bus width of adapter cards that can be inserted into this slot in bits. If the value is Unknown, enter 0. If the value is other than 8, 16, 32, 64 or 128, enter 1. It is expressed in bits.
SLOT.ThermalRating	Slot_Thermal Rating (MilliWatts)	CIM.Slot. Thermal Rating	Unsigned Integer		Maximum thermal dissipation of the slot in milliwatts.
SLOT.Description	Slot_Description	CIM.SlotDescription	String	254	The description of the adapter mounted on the slot.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
DMA.DMAChannel	DMA_DMA Channel Number	CIM.DMA. DMAChannel	Unsigned Integer		The DMA channel number.
DMA.Description	DMA_Description	CIM.DMA. Description	String	254	The name of the device using the DMA channel.
DMA.Availability	DMA_Availability	CIM.DMA. Availability	Unsigned Small Integer		Indicates whether the DMA channel is available. Enumeration values are as follows: 1 = "Other" 2 = "Unknown" 3 = "Available" 4 = "In Use/Not Available" 5 = "In Use and Available/Shareable"
DMA.BurstMode	DMA_DMA Burst Mode	CIM.DMA. BurstMode	BIT (used for Boolean condition here)		Indication that the DMA channel supports the burst mode.
NetWareOperating System.Version	NetWare.Version	ZENworks.NetWareOperating.Version	String	254	Version of the NetWare operating system.
Memory.TotalMemory	Memory_TotalMemory(MB)	ZENOperatingSystem.TotalVisibleMemorySize	Integer		Total memory of the Windows operating system.
MSDomainName.Label	WindowsDomain_Name	ManageWise.MSDomainName	String	254	The Windows domain to which the server is attached.
Monitor.DeviceID	Monitor_DeviceID	ZENworks.ZENDesktopMonitor.DeviceID	Integer		Unique ID of a desktop monitor that is attached to a computer system.
Monitor.Description	Monitor_Description	ZENworks.ZENDesktopMonitor.Description	varchar	254	Description of the monitor.
Monitor.ModelID	Monitor_ModelID	ZENworks.ZENDesktopMonitor.ModelID	varchar		Unique ID of a model of the monitor. It is a combination of the Manufacturer ID and Product ID.
Monitor.ManufactureDate	Monitor_ManufactureDate	ZENworks.ZENDesktopMonitor.ManufacturerDate	char	25	Year in which the monitor was manufactured.
Monitor.ViewableSize	Monitor_ViewableSize	ZENworks.ZENDesktopMonitor.ViewableSize	integar		A number representing the diagonal width of the screen image excluding the black borders around the image's edge.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
Monitor.NominalSize	Monitor_NominalSize	ZENworks.ZENDesktopMonitor.NominalSize	integar		A number representing the diagonal width of the monitor (the distance from one corner of the screen to the opposite corner of the screen).
Monitor.Serial Number	Monitor_Serial Number	ZENworks.ZENDesktopMonitor.Serial Number	varchar	128	Manufacturer's number used to identify a monitor.
Monitor.Manufacturer	Monitor_Manufacturer	ZENworks.ZENDesktopMonitor.Manufacturer	varchar	254	Name of the monitor's manufacturer.
Monitor.Model	Monitor_Model	ZENworks.ZENDesktopMonitor.Model	varchar	254	Product name of the monitor given by the manufacturer.
Chassis.AssetTag	Chassis_AssetTag	ZENworks_ZENChassis	varchar	254	Asset tag number of the system chassis.
Chassis.ChassisType	Chassis_ChassisType	ZENworks_ZENChassis	unsigned small int		Represents whether the system chassis is a laptop, desktop, notebook, docking station and so on.
Chassis.NumberOfPowerCords	Chassis_NumberOfPowerCords	ZENworks_ZENChassis	varchar	128	Total number of power cords attached to a system chassis.
Chassis.Manufacturer	Chassis_Manufacturer	ZENworks_ZENChassis	varchar	254	Name of the system chassis manufacturer.
Chassis.SerialNumber	Chassis_SerialNumber	ZENworks_ZENChassis	varchar	128	Manufacturer's number used to identify a system chassis.
Chassis.Version	Chassis_Version	ZENworks_ZENChassis	varchar	64	Version number of the system chassis.
Chassis.Tag	Chassis_Tag	ZENworks_ZENChassis	varchar	64	Unique ID of the system chassis attached to a particular computer system.
Software.ProductIdentifier	Software_productIdentifier	MW_DBA.InstalledSoftware.productIdentifier	varchar	254	A unique, 16-character identifier for an installed product. This identifier is available from MSI on Windows.
Software.InternalVersion	Software_InternalVersion	MW_DBA.InstalledSoftware.InternalVersion	varchar	64	Internal version of a product
Software.Language	Software_Language	MW_DBA.InstalledSoftware.Language	smallint		User-friendly name for the language of this copy of the product.
Software.UninstallString	Software_UninstallString	MW_DBA.InstalledSoftware.UninstallString	varchar	254	The command to invoke for uninstalling this product instance. Currently, this is available in Add-Remove Programs (ARP) and MSI on Windows.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
Software.InstallationSource	Software_InstallationSource	MW_DBA.InstalledSoftware.InstallationSource	varchar	254	Identifies the file system path where the installation files were stored when installing this product instance. Currently, this is available in ARP and MSI on Windows.
Software.FriendlyName	Software_FriendlyName	MW_DBA.InstalledSoftware.FriendlyName	varchar	254	Display name of the software.
Software.LastExecutionTime	Software_LastExecutionTime	MW_DBA.InstalledSoftware.LastExecutionTime	bigint		Date and time stamp when the product was last executed.
Software.FrequencyOfUsage	Software_FrequencyOfUsage	MW_DBA.InstalledSoftware.FrequencyOfUsage	smallint		Number of times the product is used.
Software.Description	Software_Description	MW_DBA.InstalledSoftware.Description	varchar	254	Description of the product.
Software.DefinitionDate	Software_DefinitionDate	MW_DBA.InstalledVirusScanner.DefinitionDate	bigint		The date of the virus definition file installed on the computer. Some anti-virus products combine date and version into a single string.
Software.DefinitionVersion	Software_DefinitionVersion	MW_DBA.InstalledVirusScanner.DefinitionVersion	varchar	64	The vendor-defined version of the virus definition file that has been installed on a computer
Software.Edition	Software_Edition	MW_DBA.ProductEdition.Name	varchar	128	Product edition defined by the vendor. For example, Professional.
Software.SupportPack	Software_Supportpack	MW_DBA.SupportPack.Name	varchar	128	Support pack name.
Software.Path	Software_Path	MW_DBA.Directory.Path	varchar	254	Directory path where the product is installed on the computer system.
Software.Name	Software_Name	MW_DBA.Software.Name	varchar	254	Vendor-defined name of the product represented as a vendor trademark or registered trademark.
Software.Vendor	Software_Vendor	MW_DBA.Software.Vendor	varchar	254	Name of the software manufacturer
Software.Version	Software_Version	MW_DBA.Software.Version	varchar	64	User-friendly version of a product.
Software.Category	Software_Category	MW_DBA.Software.Category	varchar	64	Product category to which the product belongs.
Software.HelpLink	Software_HelpLink	MW_DBA.Software.HelpLink	varchar	254	Support web site URL for the product that is available in ARP and MSI.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .csv file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
Software.PackageGUID	Software_PackageGUID	MW_DBA.Software.PackageGUID	varchar	64	Vendor-defined GUID for a product that is available in MSI.
Software.PatchName	Software_PatchName	MW_DBA.Patch.Name	varchar	254	Vendor-defined name for the patch.
File.Name	File_Name	MW_DBA.File.Name	varchar	254	Name of the file representing the software.
File.FileVersion	File_FileVersion	MW_DBA.File.FileVersion	varchar	64	Version of the file representing the software.
File.InternalName	File_InternalName	MW_DBA.File.InternalName	varchar	254	Internal name.
File.ProductVersion	File_ProductVersion	MW_DBA.File.ProductVersion	varchar	64	The version of the product represented by this file.
File.Size	File_size	MW_DBA.File.size	bigint		Size of the file representing the software.
File.LastModified	File_LastModified	MW_DBA.File.LastModified	bigint		Last modified date of the file representing the software.
File.Company	File_Company	MW_DBA.File.Company	varchar	254	Vendor name.
File.ProductName	File_ProductName	MW_DBA.File.ProductName	varchar	254	The product which this file represents.
File.Language	File_Language	MW_DBA.File.Language	smallint		User-friendly name for the language of this copy of the file
File.SoftwareDictionaryID	File_SoftwareDictionaryID	MW_DBA.File.SoftwareDictionaryID	varchar	64	ID of the file as represented in the General software dictionary.
DiskUsage.TotalDiskUsage	DiskUsage.TotalDiskUsage	MW_DBA.DiskUsage.TotalDiskUsage	bigint		Total disk usage for all the files of the specified extension.
DiskUsage.Name	DiskUsage.Name	MW_DBA.DiskUsage.Name	varchar	32	The file extension for which the disk usage is scanned for.

K Enumeration Values

This section provides information on the following topics:

Enumeration Values for General-System Information-Management Technology

1 = Unknown	3= DMI Enabled	5= SNMP Enabled
2 = Other	4 = WMI Enabled	6 = DMI and WMI Enabled

Enumeration Values for General-Inventory Information-Scan Mode

1 = Unknown	3= DMI	5= SNMP
2 = Other	4 = WMI	6 = DMI and WMI

Enumeration Values for Software-Operating Systems-Windows - Name

0 = Unknown	18 = WINNT	59 = Dedicated
1 = Other	21 = NetWare	63 = Windows (R) Me
16 = WIN95	36 = Linux	67 = Windows XP
17 = WIN98	58 = Windows	

Enumeration Values for Installation Repository

The following Installation Repository enum values are displayed in the Software > Software Group Components > Software Group and Software > Software Components > Software classes.

0 = Others	7 = MSI, Add Remove Programs, Software Dictionary	20 = Software Dictionary, Probe
1 = MSI	8 = NetWare Products.dat	21= MSI, Software Dictionary, Probes
2 = Add/Remove Programs	12 = Software Dictionary, NetWare Products.dat	22 = Add Remove programs, Software Dictionary, Probe
3 = MSI, Add Remove Programs	16 = Probe	23 = MSI, Add Remove programs, Software Dictionary, Probe
4 = Software Dictionary	17 = MSI, Probe	24 = NetWare Products.dat, Probe
5 = MSI, Software Dictionary	18 = Add Remove Programs, Probe	28 = Software Dictionary, NetWare Products.dat, Probe
6 = Add Remove Programs, Software Dictionary	19 = MSI, Add Remove Programs, Probe	

Enumeration Values for Hardware-Display Adapter-Video Architecture

1 = Other
2 = Unknown
3 = CGA
4 = EGA
5 = VGA
6 = SVGA
7 = MDA
8 = HGC
9 = MCGA
10 = 8514A
11 = XGA
12 = Linear Frame Buffer
160 = PC-98

Enumeration Values for Hardware-Display Adapter-Video Memory Type

1 = Other
2 = Unknown
3 = VRAM
4 = DRAM
5 = SRAM
6 = WRAM
7 = EDO RAM
8 = Burst Synchronous DRAM
9 = Pipelined Burst SRAM
10 = CDRAM
11 = 3DRAM
12 = SDRAM
13 = SGRAM

Enumeration Values for Hardware-Pointing Device-Name

1 = Other
2 = Unknown
3 = Mouse
4 = Track Ball
5 = Track Point
6 = Glide Point
7 = Touch Pad
8 = Touch Screen
9 = Mouse - Optical Sensor

Enumeration Values for Hardware-Battery-Chemistry

1 = Other
2 = Unknown
3 = Lead Acid
4 = Nickel Cadmium
5 = Nickel Metal Hydride
6 = Lithium-ion
7 = Zinc air
8 = Lithium Polymer

Enumeration Values for Hardware-Processor-Processor Family

1 = Other

2 = Unknown

11 = Pentium(R) Brand

12 = Pentium(R) Pro

13 = Pentium(R) II

14 = Pentium(R) Processor with MMX(TM) Technology

15 = Celeron(TM)

16 = Pentium(R) II Xeon(TM)

17 = Pentium(R) II

24 = AMD Duron(TM) Processor Family

25 = K5 Family

26 = K6 Family

27 = K6 -2

28 = K6 -3

29 = AMD Athlon (TM) Processor Family

30 = AMD29000 Family

31 = K6-2+

130 = Itanium(TM) Processor

176 = Pentium(R) III Xeon(TM)

177= Pentium(R) III Processor with Intel(R) SpeedStep(TM) Technology

178 = Pentium(R) 4 Processor

181 = Inter(R) Xeon (TM) Processor MP

182 = AMD Athlon XP (TM) Processor Family

183 = AMD Athlon MP(TM) Processor Family

300 = 6 x 86

Enumeration Values for Hardware-Processor-Upgrade Method

1= Other

2 = Unknown

3 = Daughter Board

4 = ZIF Socket

5 = Replacement/Piggy Back

6 = None

7 = LIF Socket

8 = Slot 1

9 = Slot 2

10 = 370 Pin Socket

11 = Slot A

12 = Slot M

Enumeration Values for Hardware-Chassis-Chassis Type

1 = Other

2 = Unknown

3 = Desktop

4 = Low Profile Desktop

5 = Pizza Box

6 = Mini Tower

7=Tower

8 = Portable

9 = LapTop

10 = Notebook

11 = Hand Held

12 = Docking Station

13 = All in One

14 = Sub Notebook

15 = Space-Saving

16 = Lunch Box

17 = Main System Chassis

18 = Expansion Chassis

19 = SubChassis

20 = Bus Expansion Chassis

21 = Peripheral Chassis

22 = Storage Chassis

23 = Rack Mount Chassis

24 = Sealed-Case PC

25 = Multi-system Chassis

Enumeration Values for Hardware-Bus-Protocol Supported

0 = Internal
1 = ISA
2 = EISA
3 = MicroChannel
4= TurboChannel
5 = PCI Bus
6 = VME Bus
7 = NuBus
8 = PCMCIA Bus
9 = C Bus
10 = MPI Bus
11 = MPSA Bus
12 = Internal Processor
13 = Internal Power Bus
14 = PNP ISA Bus
15 = PNP Bus
16= Maximum Interface Type

Enumeration Values for Hardware-Processor-Role

1 = Other
2 = Unknown
3= Central Processor
4 = Math Processor
5= DSP Processor
6 = Video Processor

Enumeration Values for System-System Cache-Level

1 = Other
2 = Unknown
3 = Write Back
4 = Write Through
5 = Varies with Address
6 = Determination Per I/O

Enumeration Values for System-System Cache-Cache Type

1 = Other
2 = Unknown
3 = Instruction
4 = Data
5 = Unified

Enumeration Values for System-System Cache-Replacement Policy

1 = Other
2 = Unknown
3 = Least Recently Used (LRU)
4 = First In First Out (FIFO)
5 = Last In First Out (LIFO)
6 = Least Frequently Used (LFU)
7 = Most Frequently Used (MFU)
8 = Data Dependent Multiple Algorithms

Enumeration Values for System-System Cache-Read Policy

1 = Other
2 = Unknown
3 = Read
4 = Read-ahead
5 = Read and Read-ahead
6 = Determination Per I/O

Enumeration Values for System-System Cache-Write Policy

1 = Other
2 = Unknown
3 = Write Back
4 = Write Through
5 = Varies with Address
6 = Determination Per I/O

Enumeration Values for System-System Cache-Associativity

1 = Other
2 = Unknown
3 = Direct Mapped
4 = 2-way Set-Associative
5 = 4-way Set-Associative
6 = Fully Associative
7 = 8-way Set-Associative
8 = 16-way Set-Associative

Enumeration Values for System-System IRQ-Availability

1 = Other
2 = Unknown
3 = Available
4 = In Use/Not Available
5 = In Use and Available/ Shareable

Enumeration Values for System-System IRQ-IRQ Trigger Type

1 = Other
2 = Unknown
3 = Level
4 = Edge

Enumeration Values for System-System DMA-Availability

1 = Other
2 = Unknown
3 = Available
4 = In Use/Not Available
5 = In Use and Available/ Shareable

Enumeration Values for Language

The following Language enum values are displayed in the following classes: Software Group, Software Group File Information, Software, File Information, and Exclude Information.

0=Neutral
1=Arabic
2=Bulgarian
3=Catalan
97=Not supported
101=Divehi
127=Invariant Locale
1024=Process or User Default Language
1037=Hebrew
1038=Hungarian
1081=Windows 2000/XP: Hindi. This is Unicode only.
1095=Windows XP: Gujarati. This is Unicode only.

4=Chinese

5=Czech

6=Danish

7=German

8=Greek

9=English

10=Spanish

11=Finnish

12=French

13=Hebrew

14=Hungarian

15=Icelandic

16=Italian

17=Japanese

18=Korean

19=Dutch

20=Norwegian

21=Polish

22=Portuguese

24=Romanian

25=Russian

26=Croatian

27=Slovak

28=Albanian

29=Swedish

30=Thai

31=Turkish

32=Urdu

33=Indonesian

34=Ukrainian

35=Belarusian

36=Slovenian

2048=System Default Language

1078=Afrikaans

1052=Albanian

1025=Arabic (Saudi Arabia)

2049=Arabic (Iraq)

3073=Arabic (Egypt)

4097=Arabic (Libya)

5121=Arabic (Algeria)

6145=Arabic (Morocco)

7169=Arabic (Tunisia)

8193=Arabic (Oman)

9217=Arabic (Yemen)

10241=Arabic (Syria)

11265=Arabic (Jordan)

12289=Arabic (Lebanon)

13313=Arabic (Kuwait)

14337=Arabic (U.A.E.)

15361=Arabic (Bahrain)

16385=Arabic (Qatar)

1067=Windows 2000/XP: Armenian. This is Unicode only.

1068=Azeri (Latin)

2092=Azeri (Cyrillic)

1069=Basque

1059=Belarusian

1026=Bulgarian

1109=Burmese

1027=Catalan

1028=Chinese (Taiwan)

2052=Chinese (PRC)

3076=Chinese (Hong Kong SAR, PRC)

4100=Chinese (Singapore)

5124=Windows 98/Me, Windows 2000/XP: Chinese (Macau SAR)

1039=Icelandic

1057=Indonesian

1040=Italian (Standard)

2064=Italian (Switzerland)

1041=Japanese

1099=Windows XP: Kannada. This is Unicode only.

1111=Windows 2000/XP: Konkani. This is Unicode only.

1042=Korean

2066=Windows 95

1088=Windows XP: Kyrgyz.

1062=Latvian

1063=Lithuanian

2087=Windows 98 only: Lithuanian (Classic)

1071=FYRO Macedonian

1086=Malay (Malaysian)

2110=Malay (Brunei Darussalam)

1102=Windows 2000/XP: Marathi. This is Unicode only.

1104=Windows XP: Mongolian

1044=Norwegian (Bokmal)

2068=Norwegian (Nynorsk)

1045=Polish

1046=Portuguese (Brazil)

2070=Portuguese (Portugal)

1094=Windows XP: Punjabi. This is Unicode only.

1048=Romanian

1049=Russian

1103=Windows 2000/XP: Sanskrit. This is Unicode only.

3098=Serbian (Cyrillic)

2074=Serbian (Latin)

1051=Slovak

1060=Slovenian

1034=Spanish (Spain, Traditional Sort)

37=Estonian

38=Latvian

39=Lithuanian

41=Farsi

42=Vietnamese

43=Armenian

44=Azeri

45=Basque

47=FYRO Macedonian

54=Afrikaans

55=Georgian

56=Faeroese

57=Hindi

62=Malay

63=Kazak

64=Kyrgyz

65=Swahili

67=Uzbek

68=Tatar

69=Not supported

70=Punjabi

71=Gujarati

72=Not supported

73=Tamil

74=Telugu

75=Kannada

76=Not supported

77=Not supported

78=Marathi

79=Sanskrit

1050=Croatian

1029=Czech

1030=Danish

1125=Windows XP: Divehi. This is Unicode only.

1043=Dutch (Netherlands)

2067=Dutch (Belgium)

1033=English (United States)

2057=English (United Kingdom)

3081=English (Australian)

4105=English (Canadian)

5129=English (New Zealand)

6153=English (Ireland)

7177=English (South Africa)

8201=English (Jamaica)

9225=English (Caribbean)

10249=English (Belize)

11273=English (Trinidad)

12297=Windows 98/Me, Windows 2000/XP: English (Zimbabwe)

13321=Windows 98/Me, Windows 2000/XP: English (Philippines)

1061=Estonian

1080=Faeroese

1065=Farsi

1035=Finnish

1036=French (Standard)

2060=French (Belgian)

3084=French (Canadian)

4108=French (Switzerland)

5132=French (Luxembourg)

6156=Windows 98/Me, Windows 2000/XP: French (Monaco)

1110=Windows XP: Galician

2058=Spanish (Mexican)

3082=Spanish (Spain, Modern Sort)

4106=Spanish (Guatemala)

5130=Spanish (Costa Rica)

6154=Spanish (Panama)

7178=Spanish (Dominican Republic)

8202=Spanish (Venezuela)

9226=Spanish (Colombia)

10250=Spanish (Peru)

11274=Spanish (Argentina)

12298=Spanish (Ecuador)

13322=Spanish (Chile)

14346=Spanish (Uruguay)

15370=Spanish (Paraguay)

16394=Spanish (Bolivia)

17418=Spanish (El Salvador)

18442=Spanish (Honduras)

19466=Spanish (Nicaragua)

20490=Spanish (Puerto Rico)

1072=Sutu

1089=Swahili (Kenya)

1053=Swedish

2077=Swedish (Finland)

1114=Windows XP: Syriac. This is Unicode only.

1097=Windows 2000/XP: Tamil. This is Unicode only.

1092=Tatar (Tatarstan)

1098=Windows XP: Telugu. This is Unicode only.

1054=Thai

1055=Turkish

1058=Ukrainian

80=Mongolian	1079=Windows 2000/XP: Georgian. This is Unicode only.	1056=Windows 98/Me, Windows 2000/XP: Urdu (Pakistan)
86=Galician	1031=German (Standard)	2080=Urdu (India)
87=Konkani	2055=German (Switzerland)	1091=Uzbek (Latin)
88=Not supported	3079=German (Austria)	2115=Uzbek (Cyrillic)
89=Not supported	4103=German (Luxembourg)	1066=Windows 98/Me, Windows NT 4.0 and later: Vietnamese
90=Syriac	5127=German (Liechtenstein)	
96=Not supported	1032=Greek	

L Documentation Updates

This section contains information on documentation content changes that have been made in the *Administration* guide for Server Inventory since the initial release of Novell® ZENworks® 6.5 Server Management. The information will help you to keep current on updates to the documentation.

All changes that are noted in this section were also made in the documentation. The documentation is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the documentation changes listed in this section.

The documentation update information is grouped according to the date the changes were published. Within a dated section, the changes are alphabetically listed by the names of the main table of contents sections for Server Inventory.

If you need to know whether a copy of the PDF documentation you are using is the most recent, the PDF document contains the date it was published on the front title page or in the Legal Notices section immediately following the title page.

The documentation was updated on the following dates:

December 23, 2005

Updates were made to the following sections.

Location	Change
"Configuring the MS SQL Server 2000 Inventory Database" on page 479.	Added the following note in Step 11e on page 480: NOTE: During the execution of the drop trigger sqls, the following error message might be displayed on the console, "Cannot drop the trigger '*trigger_name*', because it does not exist in the system catalog". Ignore the error message.
"Generating Inventory Reports" on page 642.	Added the following note in Step 4 on page 643: NOTE: ZENworks Inventory report supports only the following double-byte character languages: German, English, Spanish, French, Portugese, and Japanese. Other double-byte characters might not be displayed properly in the Inventory reports.

September 23, 2005 (Support Pack 2)

Updates were made to the following sections for ZENworks 6.5 Support Pack 2 (SP2):

Location	Change
"Backing Up the Sybase Inventory Database" on page 469	This section has been reorganized. There is no change in the content of the section.
"Creating the Oracle9i Inventory Database on a UNIX Server" on page 475	This is a new section.
"Setting Up the Oracle Inventory Database" on page 471	This section has been reorganized. There is no change in the content of the section.

June 17, 2005

Updates were made to the following sections:

Location	Change
"Manually Creating the Sybase Inventory Database Object" on page 466	Added the following important note in Step 3c on page 466: IMPORTANT: All Inventory components use the username and the password configured in the database object. By default, "novell" is the password for all options. But you can change it in the database, and update the same here.

Location	Change
“Creating the Oracle8i Inventory Database on a Windows Server” on page 471	Added a step, Step 2 on page 471.
“Creating the Oracle9i Inventory Database on a Windows Server” on page 473	Added a step, Step 2 on page 473.
“Manually Creating the Oracle Inventory Database Object” on page 476	Added the following important note in Step 3c on page 477: IMPORTANT: All Inventory components use the username and the password configured in the database object. By default, “novell” is the password for all options. But you can change it in the database, and update the same here.
“Manually Creating the MS SQL Server 2000 Inventory Database Object” on page 481	This is a new section.
“Disabling File Scan” on page 610	This is a new section.

March 11, 2005

Updates were made to the following section:

Location	Change
“Removing the Redundant Inventoried Servers from the Inventory Database” on page 616	Documented a sample inventoryremovallist.txt file.

February 11, 2005 (Support Pack 1)

Updates were made to the following sections for ZENworks 6.5 Support Pack 1 (SP1):

Location	Change
"Customizing the Software Inventory Information To Be Scanned For the ZENworks 6.5 or ZENworks 6.5 Support Packs Inventoried Servers" on page 550	Updated this section with many changes for ZENworks 6.5 Server Management Support Pack 1.
"Removing the Redundant Inventoried Servers from the Inventory Database" on page 616	Updated this section.
"Retrieving Inventory information from the Inventory Database Without Using the CIM Schema" on page 653	This is a new section.

November 10, 2004

Updates were made to the following sections:

Location	Change
"Understanding the Inventory Server Roles" on page 420	In the following sections, added the criteria to select a specific Inventory server role: ◆ "Root Server" on page 421 ◆ "Root Server with Inventoried Servers" on page 422 ◆ "Intermediate Server" on page 423 ◆ "Intermediate Server with Database" on page 424 ◆ "Intermediate Server with Inventoried Servers" on page 425 ◆ "Intermediate Server with Database and Inventoried Servers" on page 426 ◆ "Leaf Server" on page 427 ◆ "Leaf Server with Database" on page 428 ◆ "Standalone Server" on page 429
"Simple Deployment" on page 432	1. Renamed the section, "Typical Deployment - Deploying Inventory in a Single Site" to "Simple Deployment". 2. Updated this section with information to help you deploy Server Inventory in a simple deployment scenario.
"Advanced Deployment" on page 435	1. Renamed the section, "Custom Deployment - Deploying Inventory in a Multiple or Enterprise Sites" to "Advanced Deployment". 2. Updated this section with information to help you deploy Server Inventory in advanced deployment scenarios.
"Understanding the Effects of Server Inventory Installation" on page 452	Specified the rights granted to the scandir and dictdir directories.

September 2, 2004

Throughout the guide, graphics have been updated with font changes and newer icons. No notable content changes were made to any graphics.

Updates were made to the following sections:

Location	Change
"Connecting the Inventory Server and ConsoleOne to the MS SQL Server 2000 Inventory Database" on page 482	Rectified the Microsoft SQL Server web site URL to be referred for downloading the Windows English version of Microsoft JDBC driver.

July 28, 2004

Updates were made to the following sections:

Location	Change
"Understanding the Inventory Service Manager" on page 493 > "List of Services" on page 493	Appended this section with the following information: The Inventory Service Manager reads the server property file (config.properties) and the role-based property file in the *inventory_server_installation_directory_or_volume*\zenworks\inv\server\wminv\ properties directory, and loads the required services and server components. IMPORTANT: Do not modify the property files because the updates might fail to load the services or the Service Manager.
"Viewing the Inventory Servers Deployed for Inventory" on page 545	Added the following note: NOTE: You cannot collapse the inventory tree using the short-cut keys.
"Generating Inventory Reports" on page 642	Added the following guideline that must be followed as you work with the Reporting dialog box: Double-byte characters are not displayed properly in the Inventory reports.