III

Server Inventory

Novell® ZENworks® for Servers (ZfS) Server Inventory enables you to collect hardware and software inventory information from the local and the remote servers of your enterprise. This inventory information is scanned and stored in a database that can be accessed by the network administrator.

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

The Server Inventory documentation contains the following sections:

25 Understanding Server Inventory

Novell® ZENworks® for Servers (ZfS) Server Inventory gathers hardware and software inventory information from the NetWare® 5.1/6 and Windows* NT* 4.0/2000 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.

This chapter provides a basic overview of the ZfS Server Inventory service. 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 data 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 database: A repository of inventory information of all the inventoried servers.

Inventory server: A server where you run the Inventory service. This server can run any other ZfS 3 services also. The Inventory server collects the inventory data 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.

Database server: A server running Sybase* or Oracle* 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 ZfS 3 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. There is no roll-up of the inventory information.

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 the Inventory database attached to. 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 staging Inventory server for moving the data from the lower-level Inventory servers up the Inventory server hierarchy. This server can have inventoried servers or the Inventory database attached to it.

Root Server: The highest-level Inventory server in the inventory tree hierarchy. This server has a centralized 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.

Inventory site: A single site with a simple network environment of inventoried servers and at least one Inventory server. A site is typically a geographical location. There can be multiple sites your enterprise.

Overview of Server Inventory Components

Before setting up the ZfS inventory deployment, you should understand the inventory components that interact together to perform inventory functions.

ZfS Server Inventory uses the following components:

- "Inventory Scanners" on page 654
- "Inventory Components on Inventory Servers" on page 654
- "Inventory Database" on page 655
- "Management Console" on page 655

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 and store the scan data as .str files. The .str files are subsequently transferred to the Inventory server and processed.

Using the Server Inventory policy, you can configure the scan settings so you can schedule the scanning on the inventoried servers, enable a software scan, and customize software scanning. From the Inventory Service object, you can specify the location of the scan data files.

Inventory Components on Inventory Servers

The inventory components process the scan data. The following components are Java* programs that work identically on NetWare and Windows NT/2000 Inventory servers:

- Scan Collector

  The Scan Collector collects the .str files and stores them in the scan directory (scandir) at the Inventory server. The .str files are transferred using the XML-RPC protocol.

- Selector

  The Selector processes the .str files and places the files in the dbdir and entmergedir directories.

- Sender and Receiver

  The Sender and the Receiver on the Inventory servers compress the .str files and then transfer the files from the lower-level Inventory servers to the higher-level Inventory servers for roll-up of inventory information. 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.

- Storer

  The Storer stores the collected inventory information (.str files) in the Inventory database. By using the Database Location policy, you can configure the properties of the Inventory database object in ZfS and associate the database object to an Inventory server.

Inventory Database

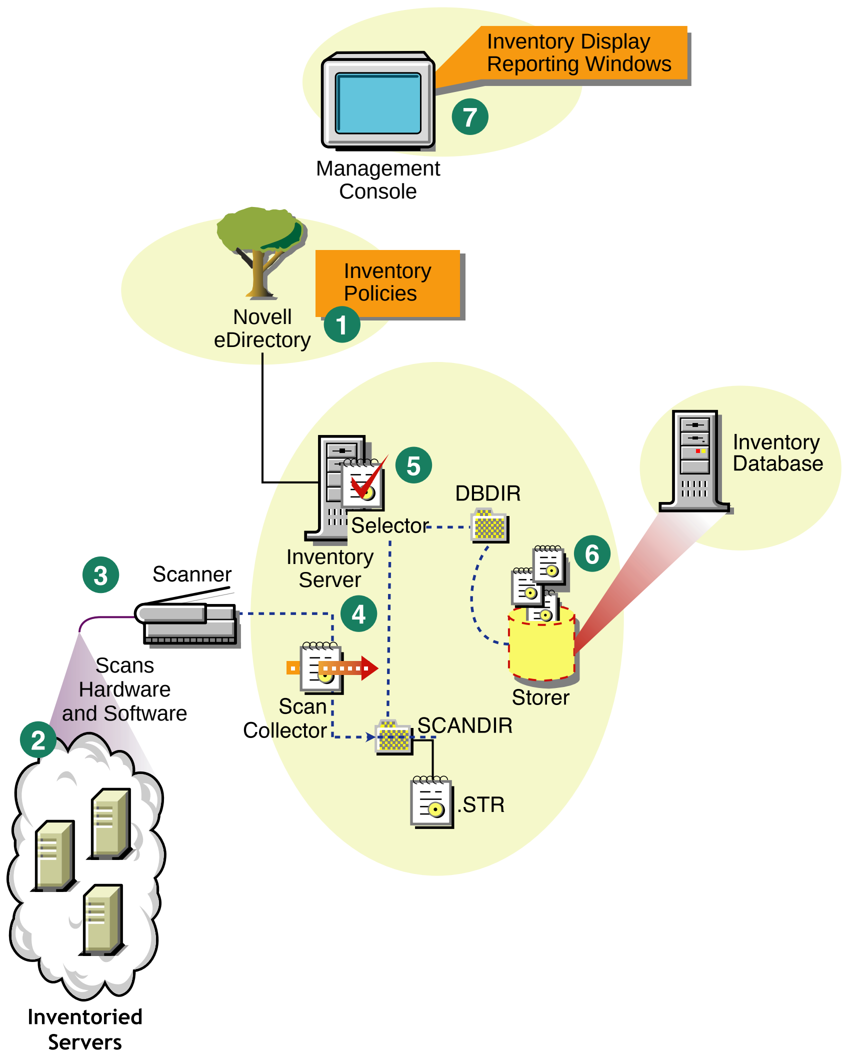
The Inventory database is a repository of inventory information of the inventoried servers. In ZfS, the database is a Common Information Model-based database but it is implemented in relational database management system (RDBMS) and maintained in Sybase* or Oracle*.

Management Console

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

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:



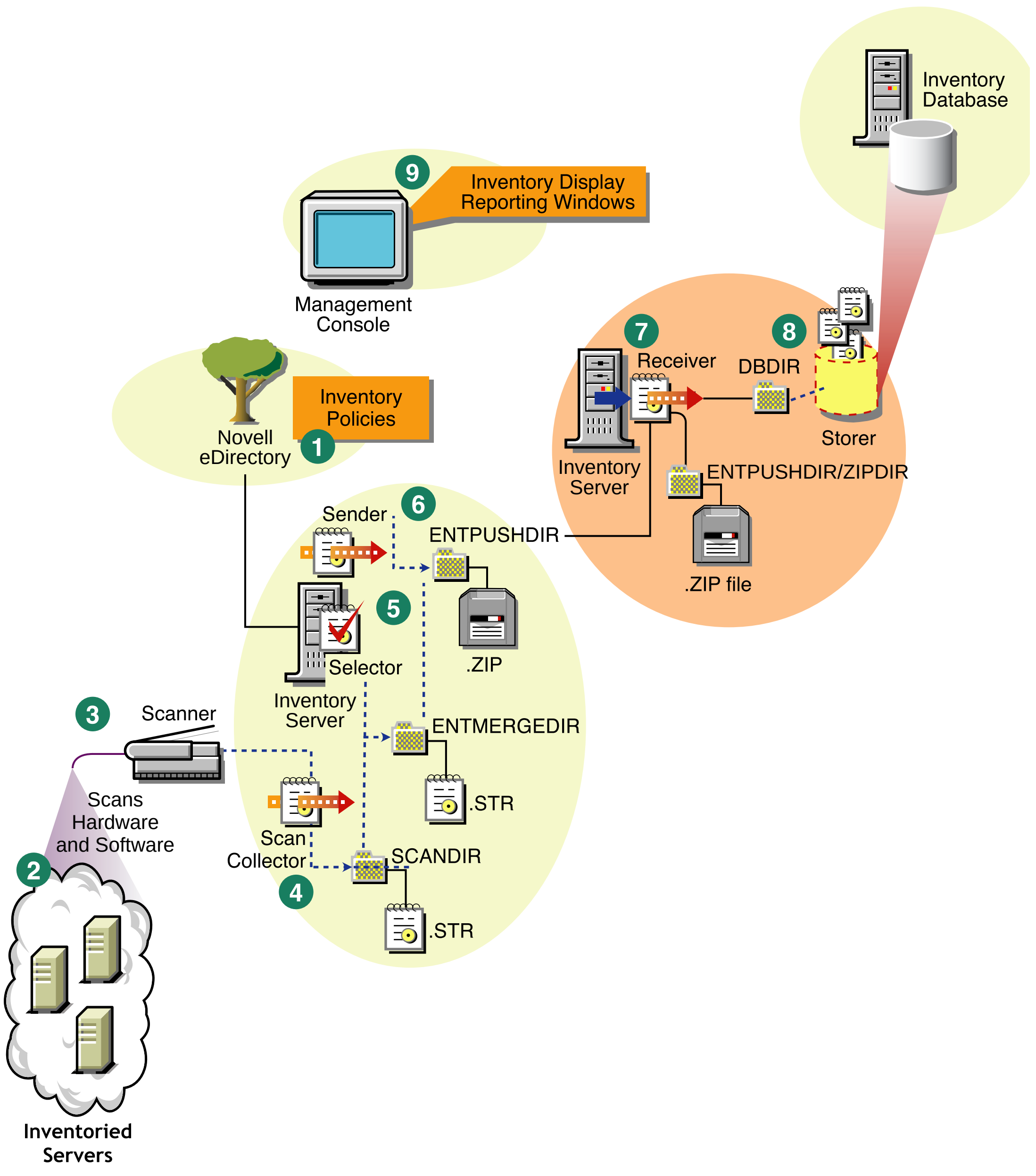
The inventory scanning cycle is as follows:

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 scan data will be sent, scanning time, whether to include software scanning of inventoried servers, 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.
3. The Scanner stores the scan data (.str) locally on the inventoried server. This data is transferred to the Inventory server using the XML-RPC protocol.
4. The Scan Collector receives the .str file using the XML-RPC protocol and stores the str file in the scan directory (scandir) at the Inventory server. The Scan Collector uses the ZEN Web Server to process the XML-RPC requests.

5. The Selector validates the .str file and places the file in the Database directory (dbdir).
6. The Storer updates the database with the inventory information of the .str file.
7. The network administrator views the inventory information and queries the database in ConsoleOne.

Understanding Rolling Up Scan Data Across Servers

The following illustration depicts rolling up the scan data across servers, which is explained below:



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

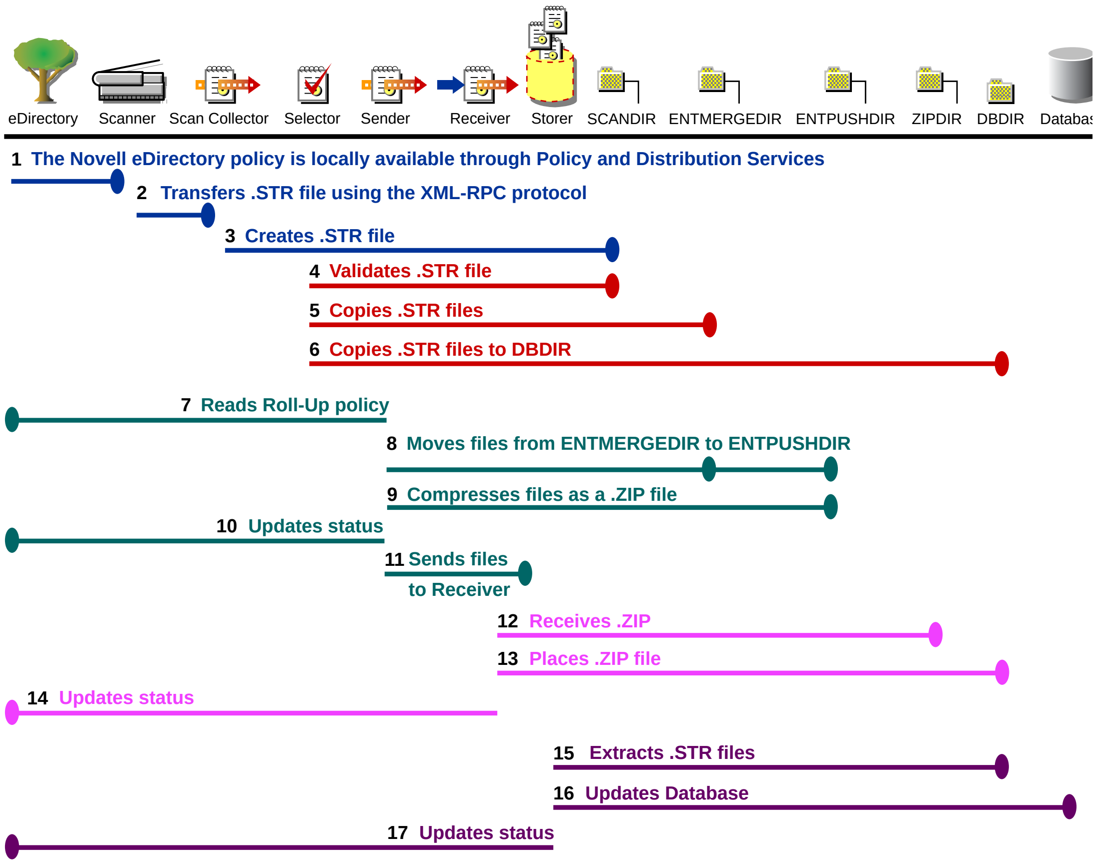
1. The inventory policies in eDirectory define the inventory settings, such as the Inventory Service object name of the Inventory server to which the scan data will be sent, scanning time, whether to include software scanning of inventoried servers, and the software rules for software scan. These settings 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.
3. The Scanner stores the scanned data (.str) locally on the inventoried server. This data is transferred to the Inventory server using the XML-RPC protocol.
4. The Scan Collector receives the .str file using the XML-RPC protocol and stores the .str file in the scan directory (scandir) at the Inventory server. The Scan Collector uses the ZEN Web Server to process the XML-RPC requests.
5. The Selector validates the .str file and places the file in the enterprise merge directory (entmergedir) for roll-up of scan data. If there is a database attached, the Selector also places the files in the database directory (dbdir).
6. The Sender on the Inventory server has a Roll-Up policy to identify the Inventory server to which it will transmit the scan data and the Roll-Up schedule specifies the time for roll-up of data. The Sender compresses the .str files as a .zip file and places the .zip file in the enterprise push directory (entpushdir). 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 directory 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 network administrator views the inventory information, and queries the database in ConsoleOne.

The following illustration lists the sequence of scan operations done by each Inventory component:



26 Setting Up Server Inventory

Before you install Novell® ZENworks® for Servers (ZfS) Server Inventory in your working environment, you must plan and decide the Inventory server tree hierarchy for your company. You should organize your inventory deployment based on your network and information requirements.

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

1. "Understanding the Inventory Server Roles" on page 661
2. "Deploying Server Inventory" on page 670
3. "Understanding the Effects of Server Inventory Installation" on page 683
4. "Setting Up the Inventory Database" on page 684
5. "Configuring Inventory Servers for Server Inventory" on page 705
6. "Starting and Stopping the Inventory Service" on page 712

You can change to role of the Inventory server. For more information, see "Changing the Role of the Inventory Server" on page 713.

Understanding the Inventory Server Roles

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

- "Root Server" on page 661
- "Root Server with Inventoried Servers" on page 663
- "Leaf Server" on page 668
- "Leaf Server with Database" on page 669
- "Intermediate Server" on page 664
- "Intermediate Server with Database" on page 665
- "Intermediate Server with Database and Inventoried Servers" on page 667
- "Standalone Server" on page 670

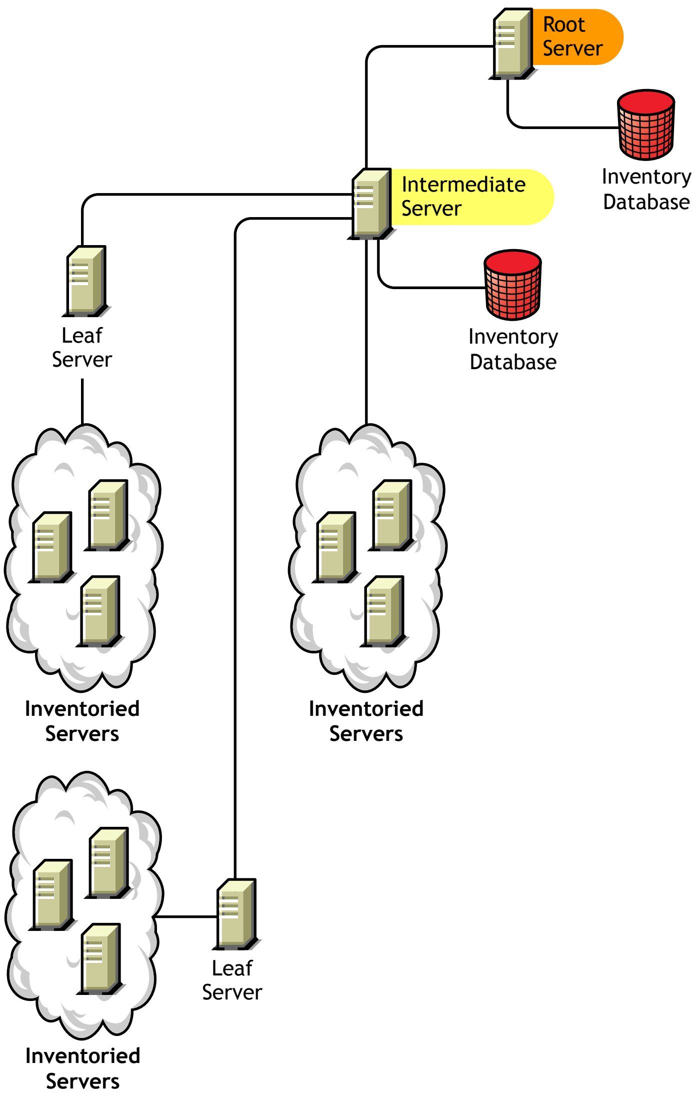
Root Server

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.

The Inventory database at the Root Server contains the inventory information for all the lower-level Inventory servers. At the Root Server level, you can view complete inventory information.

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

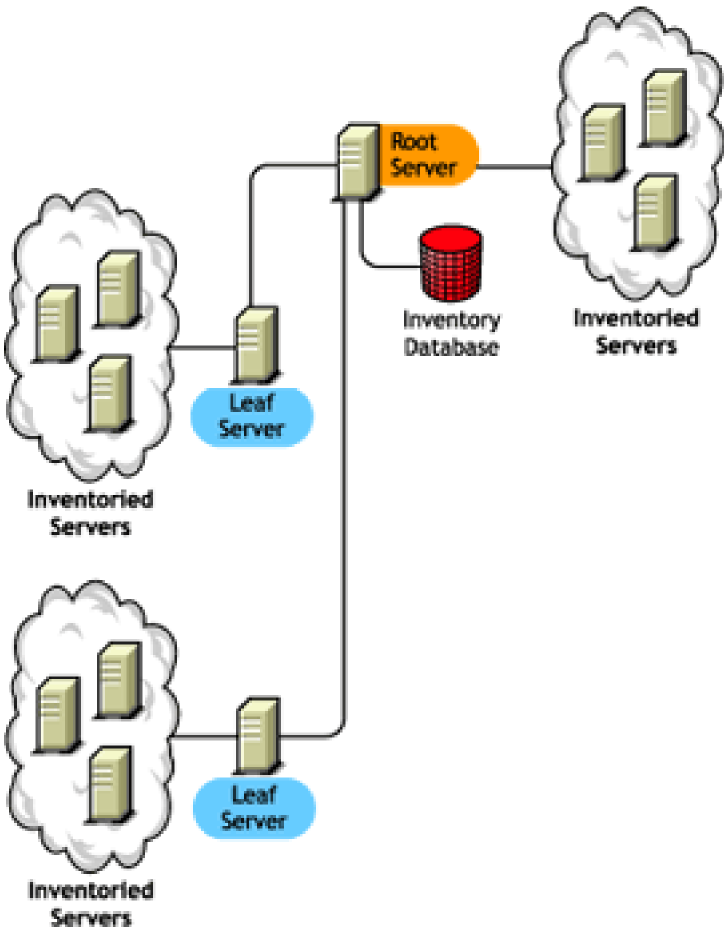


Root Server with Inventoried Servers

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 inventoried servers attached to it. There are inventoried servers residing on a LAN.
- This server has an Inventory database attached to it.

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:

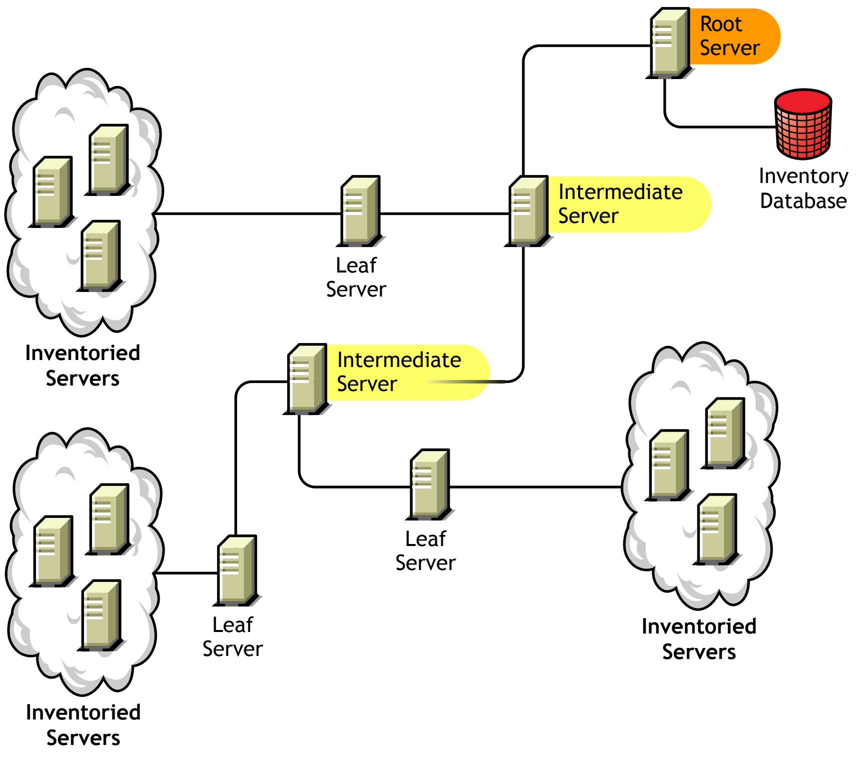


Intermediate Server

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 scan information to the next-level Inventory server or to the Root Server.
- This server does not have inventoried servers or an Inventory database attached to it.
- There can be one or more Intermediate Servers.

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.



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 scan data to the Intermediate Server.

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

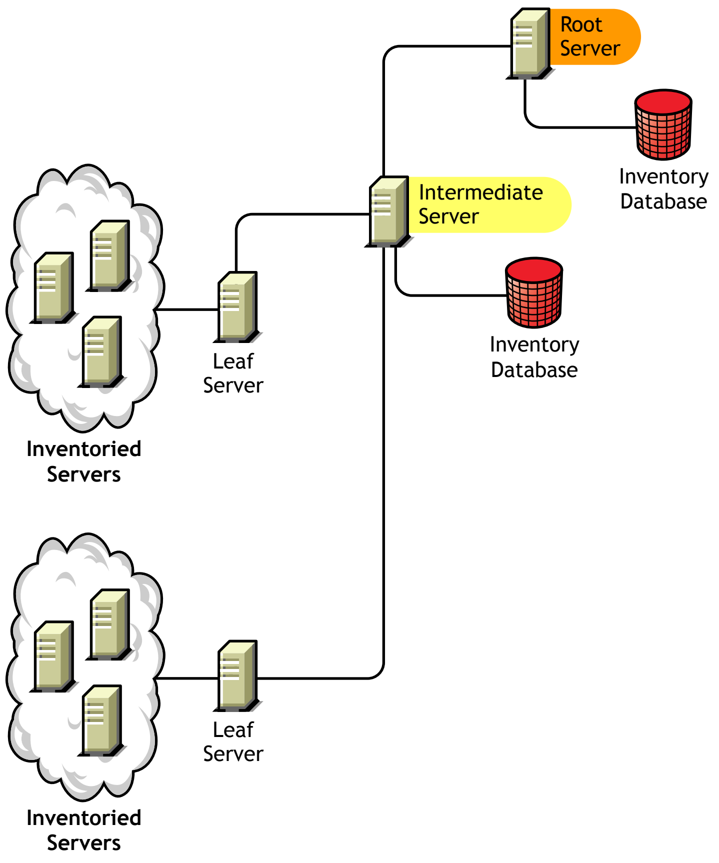
Intermediate Server with Database

The Intermediate Server with Database has the following characteristics:

- This server acts as a staging server for the lower-level Leaf Servers.
- This Inventory server moves the scan information to the next-level Intermediate Server or the Root Server.
- This server has an Inventory database attached to it.

There can be one or more Intermediate Servers in your enterprise.

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.



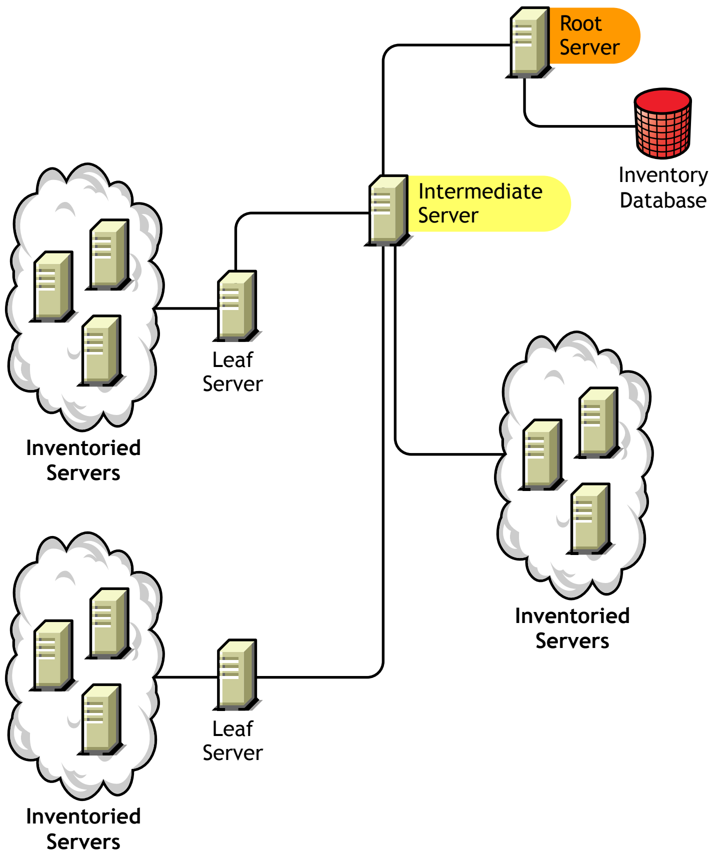
Intermediate Server with Inventoried Servers

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 scan information to the next-level Intermediate Server or to the Root 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 in your enterprise.

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

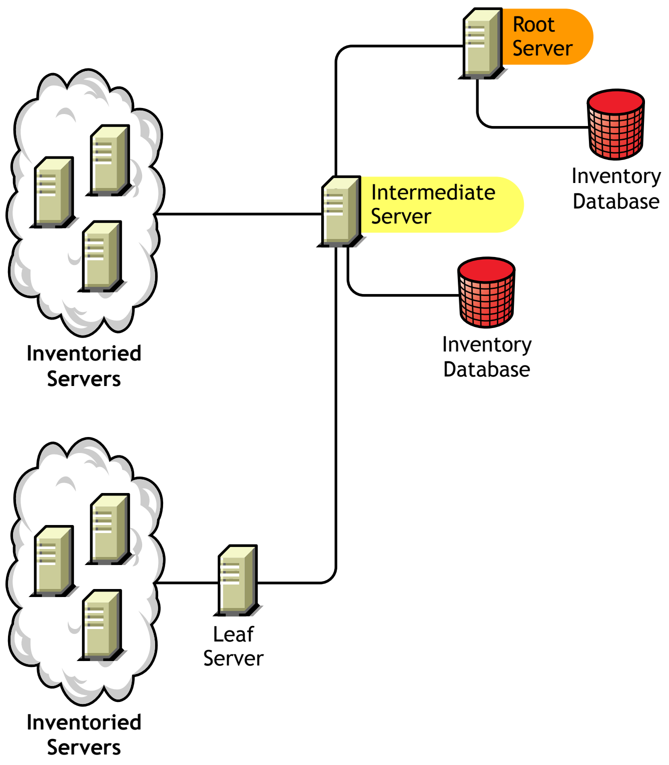


Intermediate Server with Database and Inventoried Servers

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

- This Inventory server acts as an intermediate server for the lower-level Leaf Servers.
- This server moves the scan information to the next-level Intermediate Server or to the Root Server.
- This server has inventoried servers attached to it.
- This server has Inventory database attached to it.

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.

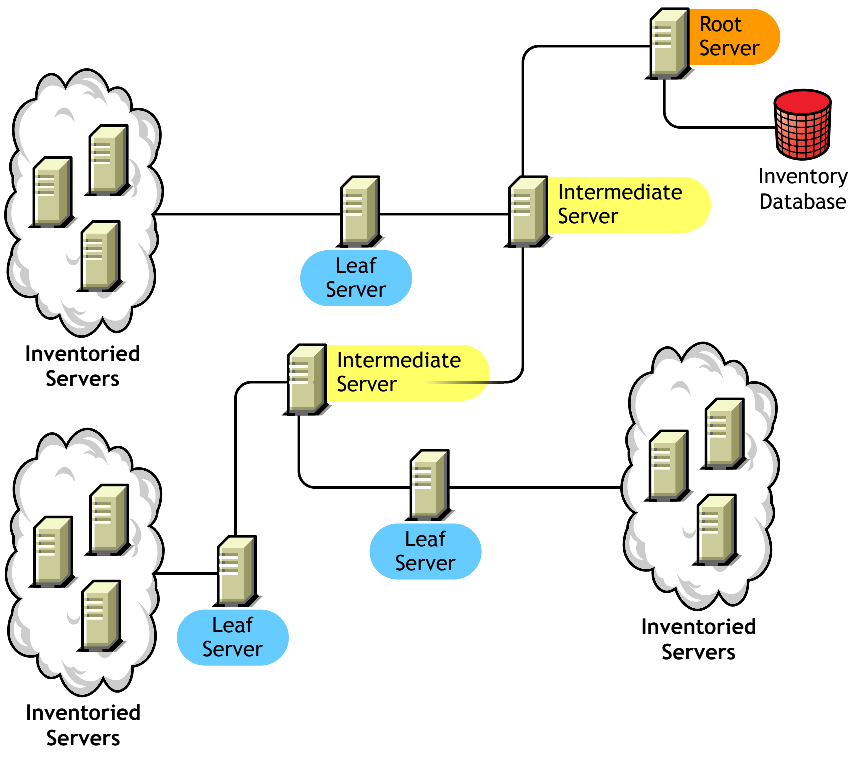


Leaf Server

The Leaf Server has the following characteristics:

- This Inventory server is at the lowest level in the hierarchy.
- This server has inventoried servers attached to it.
- This server moves the scan data to the next-level Intermediate Server or to a Root Server.
- A simple Leaf Server does not have an Inventory database. An Inventory database is not required because there may be only few inventoried servers connected to the Inventory server.

  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.

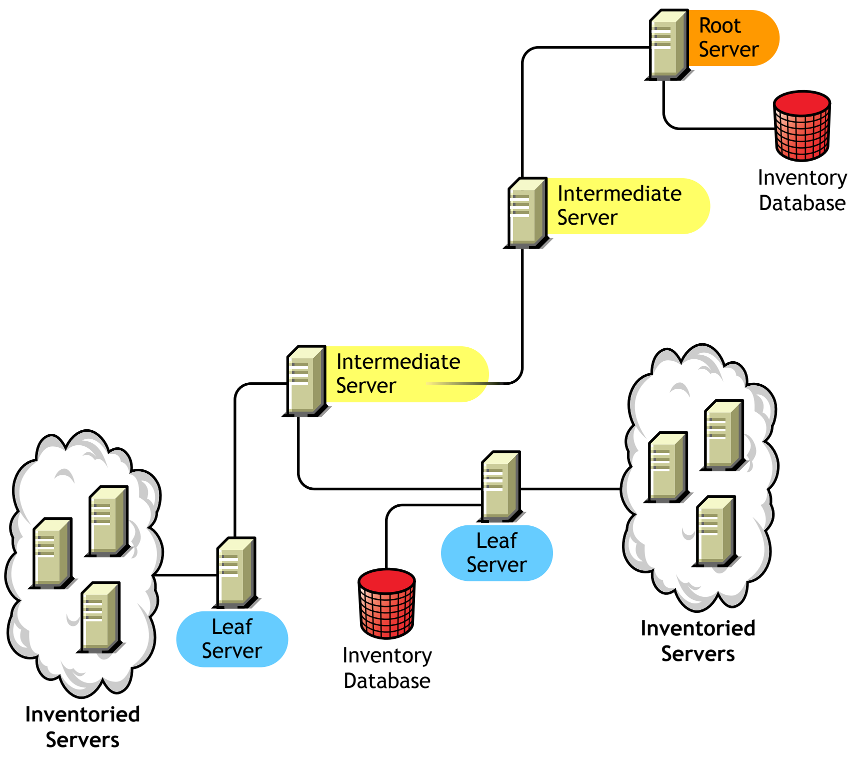


Leaf Server with Database

The Leaf Server with Database has the following characteristics:

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

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.

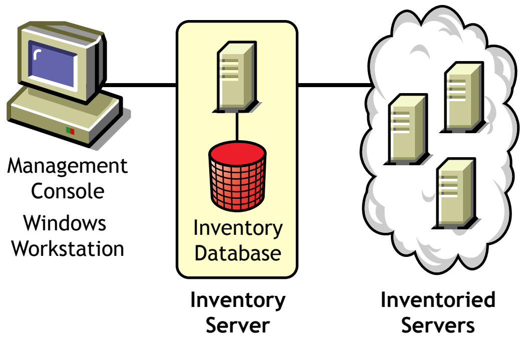


Standalone Server

The Standalone Server has the following characteristics:

- This 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 the Root Server.

The following illustration depicts Standalone Server.



Deploying Server Inventory

The following sections will help you to deploy Server Inventory:

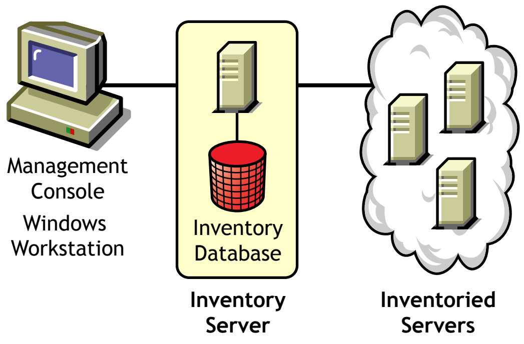
- "Deploying Server Inventory in a LAN Environment" on page 670
- "Deploying Inventory over a WAN Environment" on page 671

IMPORTANT: The recommendations discussed in the scenarios are generic. Because of the unique nature of your topology, further refinements may become necessary.

Deploying Server Inventory in a LAN Environment

In ZfS, the deployment of Server Inventory in a LAN environment implies deploying the product on a single inventory site.

In this type of inventory configuration, the Inventory server components and database are located on a Standalone Server. There is no roll-up of data and the Sender-Receiver components are not used. This scenario is illustrated in the following figure.



Recommendations for Deployment in a LAN Environment

- The minimum base Inventory server configuration includes 256 MB RAM and a database cache of 64 MB. For a higher inventoried server range, the Inventory server configuration is 512 MB RAM and a database cache of 128 MB.
- All inventoried servers should send the scan data to the nearest Inventory server on the LAN; policies must be created based on this information.
- The transmission of scan data from inventory servers can take several hours or even more than a day. The scanning is an ongoing background process.
- 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 server File System Services.
- Ensure that the time synchronization radius is set within 2 seconds.
- For all databases, the optimal database cache size requirement for the server may 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 32 MB.

Deploying Inventory over a WAN Environment

In a WAN environment, complete the following tasks, in order, to design the inventory tree and deploy inventory:

- "1. List the sites in the enterprise" on page 671
- "2. What is the ideal place for the Root Server?" on page 672
- "3. Is any other database needed?" on page 673

  "Optional step: If another database is needed" on page 673
- "4. Identify the route for Inventory data" on page 673
- "5. Identify servers on each site for Inventory, Intermediate and Database Servers" on page 674
- "6. Identify the location of the Distributors" on page 675
- "7. Create the tree of servers for company Inventory collection" on page 675
- "8. Create an implementation plan" on page 675
- "9. Start the actual deployment" on page 675

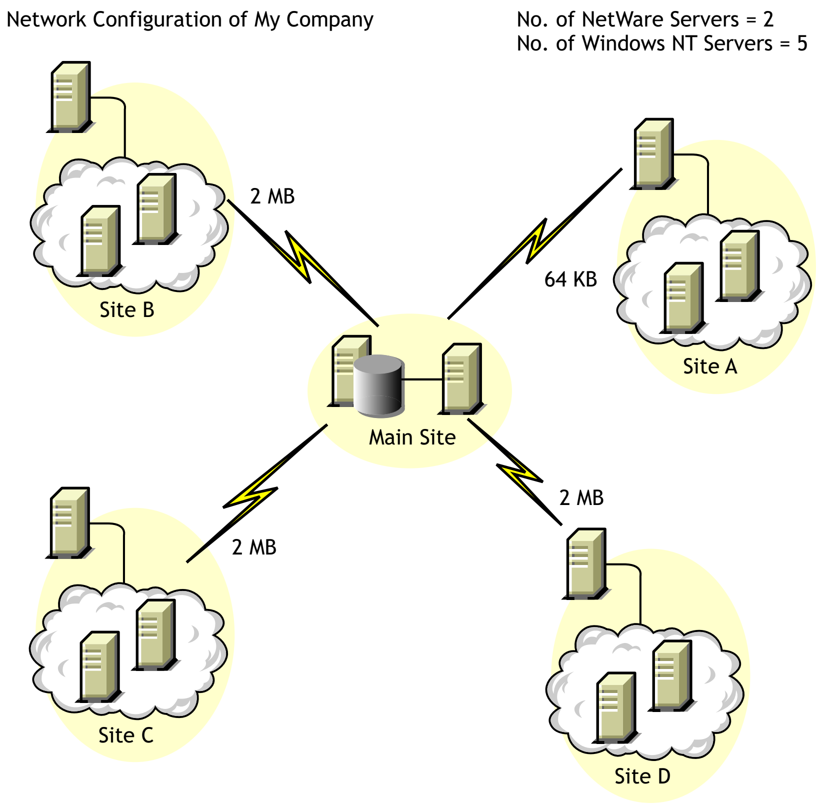
"Guidelines for Sending Inventory Information in a WAN" on page 683 covers recommendations for deployment.

1. List the sites in the enterprise

Describe the entire network of your company.

- List the various sites in your company.
- 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 company with servers in different locations.


Network Configuration of My Company
No. of NetWare Servers = 2
No. of Windows NT Servers = 5
2 MB
64 KB
Site B
Site A
Main Site
2 MB
2 MB
Site C
Site D

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.
- There must 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?

Besides the database at the Root Server, you can maintain database servers at different sites.

You may 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 may be some organizational needs for your company 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 may 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 (subtrees). Also, identify the sites that require data 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 data

Identify the routes for inventory data for all sites to the nearest database, and then identify the route to the database on the Root Server.

To devise a route plan:

- Each route can have an Intermediate Server at a staging site. The Intermediate Server receives and transmits the data to the next destination. These are application-layer-level routes for inventory data. 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 data travels from a particular site to its final destination, which is the database at the Root Server.
- There may 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 may be some cost in terms of management and traffic generation. If there is no intermediate database involved, you can change the route by only 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 different scheduling.
- Availability of Inventory servers on the intermediate site for staging the inventory data should be considered in deciding the sites for Intermediate Servers. Provide enough disk space on these servers to store all the inventory data on the disk until the Sender sends it to the next destination.
- Inventoried servers should not be connected to the inventory server over a WAN because the inventoried server scanning should not be done across a WAN.

5. Identify servers on each site for Inventory, Intermediate and Database Servers

A single server can have different roles if it has sufficient resources. For example, an Inventory server can be a Leaf Server with Database. You can also designate an Inventory server as an Intermediate Server with Database, which receives inventory from the inventoried servers and also has an Inventory database. An Inventory server can have any combination of roles.

In ZfS, you choose the role for each Inventory server. For more information, see "Understanding the Inventory Server Roles" on page 661.

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 data collected. Calculate 50 to 60 KB scan data from each inventoried server to calculate the load.

dbg is the storage space of the scan data in the database. Calculate 100 to 120 KB per inventoried server as the disk space for 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. Calculate 7 to 10 KB for the roll-up of 50 KB scan data.

{ } 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 scan data, including the database disk space.

6. Identify the location of the Distributors

The ZfS 3 Distributor component is required to distribute the inventory policies among the inventoried servers. For more information, see Chapter 14, "Configuring Policy and Distribution Services," on page 317.

7. Create the tree of servers for company 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.

8. 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.

9. 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.
2. Create the policies applicable to inventoried servers.
3. Create the Roll-Up policies to schedule the roll-up for each Inventory server.

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. See "Setting Up the Inventory Database for Oracle8i and Oracle9i" on page 690.

2 Shut down the Inventory services. For more information, see "Stopping the Inventory Service" on page 713.

3 Based on the database you select, make sure that you configure the database. See "Configuring the Database Location Policy" on page 710.

4 Modify the role of the existing Inventory server in the Inventory Service object.

If you are adding a new Inventory server, you need not modify the role of that server. If you want to change the role of the Inventory server, for example, from Leaf Server to Leaf Server with Database, you need to modify the role of the Inventory server in the Inventory Service object.

4a In ConsoleOne®, right-click the Inventory Service object (*servername*_ZenInvservice), click Properties, then click the Inventory Service Object Properties tab.

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

You will see a list of actions that you should follow based on the chosen role. For example, if you change the Root Server to Root Server with Inventoried Servers, you need to configure the Server Inventory policy for the inventoried servers that you have attached. Similarly, to change the role to any other Inventory server, you need to follow the instructions to make the role change effective.

Follow the actions that you need to change the role. For more information, see "Changing the Role of the Inventory Server" on page 713.

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

5a In ConsoleOne, right-click the Inventory Service object (*servername*_ZenInvservice), click Properties, then click the Inventory Service Object Properties tab.

5b Check the Enforce Full Scan option, then click OK.

6 Bring up the Inventory service.

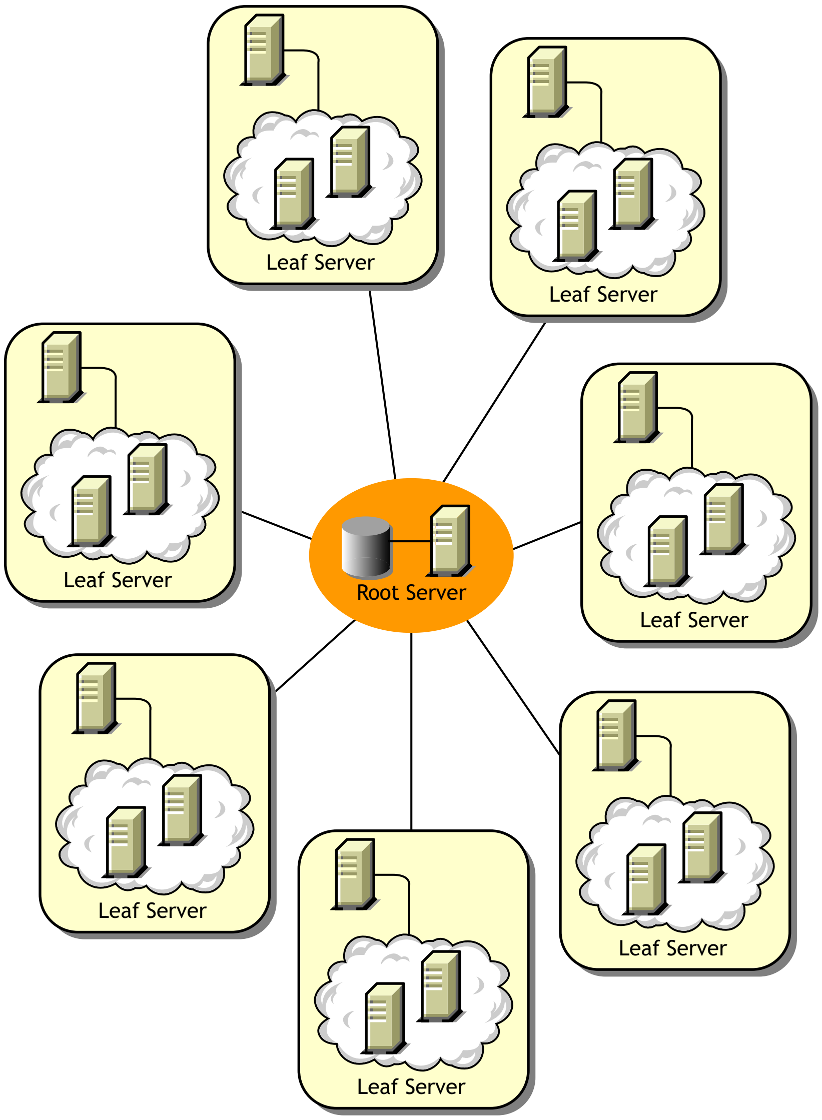
Possible Inventory Server Configurations for a WAN

The following sections cover these scenarios:

- "Scenario 1: WAN Inventory Deployment for up to 50 Inventory Sites without Intermediate Servers" on page 677
- "Scenario 2: Up to 50 Intermediate Servers Connected to the Root Server" on page 678
- "Scenario 3: Intermediate Servers with Database Connected to the Root Server" on page 679
- "Scenario 4: Database on Inventory Servers and Intermediate Servers Connected to a Root Server" on page 680
- "Scenario 5: Roll-Up of the Inventory information Across eDirectory Trees" on page 681
- "Scenario 6: Merging eDirectory Trees" on page 682
- "Scenario 7: Deploying Inventory Server Across Firewall" on page 682
- "Guidelines for Sending Inventory Information in a WAN" on page 683

Scenario 1: WAN Inventory Deployment for up to 50 Inventory Sites without Intermediate Servers

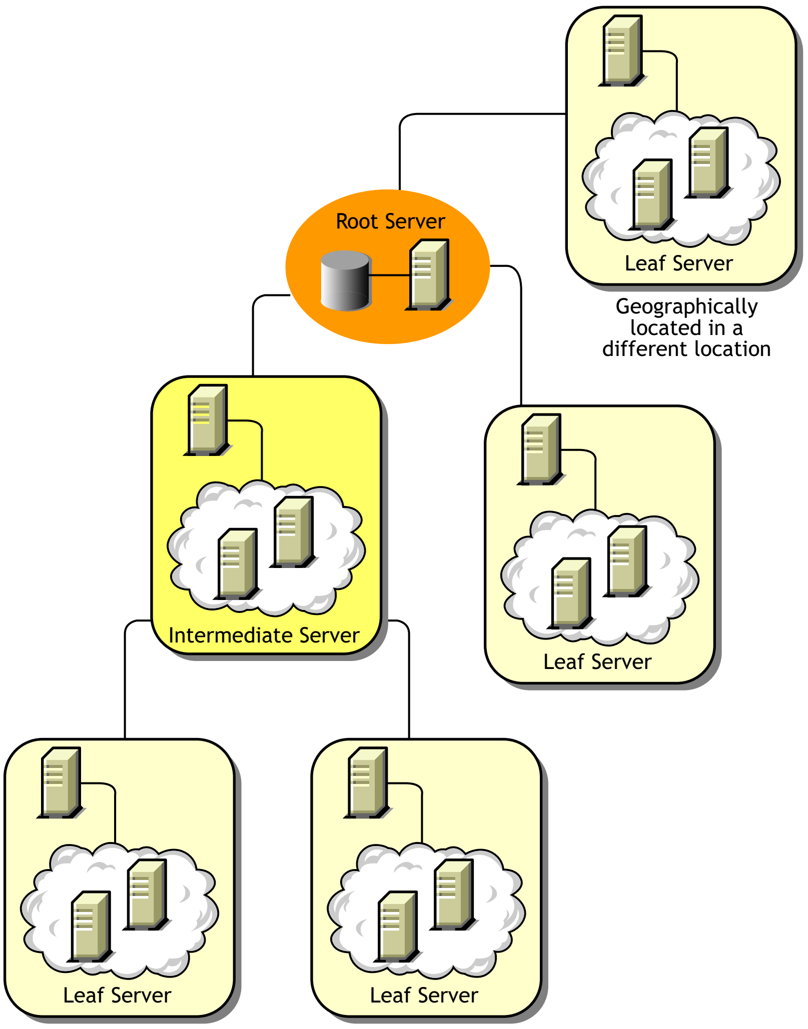
In this configuration, all Inventory servers are connected to a central enterprise database server. The Leaf Servers do not have a database and Intermediate Servers are not required. This scenario is illustrated in the following figure:



Scenario 2: Up to 50 Intermediate Servers Connected to the Root Server

In this configuration, the Leaf Servers roll up data to the next-level Intermediate Server and finally to the Root Server. Another Inventory server, at a different location, is also connected to the Root Server.

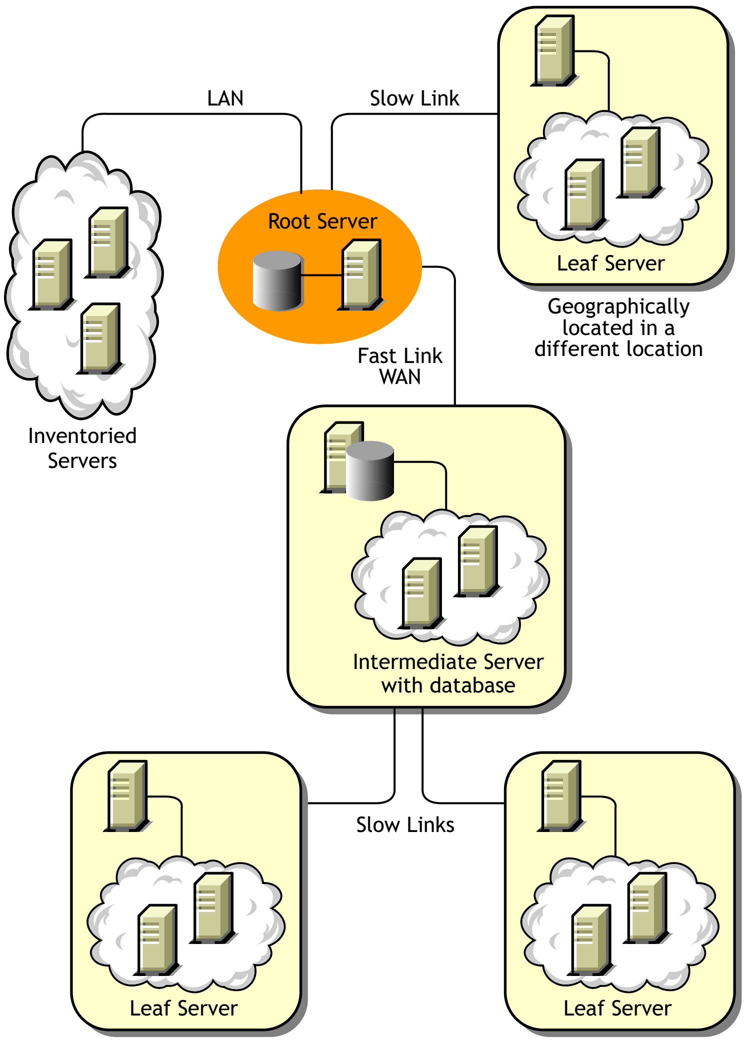
This scenario is illustrated in the following figure:



Scenario 3: Intermediate Servers with Database Connected to the Root Server

In this configuration, the inventory servers are connected to the Intermediate Server over fast WAN links. The Intermediate Server also has an Inventory database and transmits the information to the Root Server. Other Inventory servers are also connected to the Root Server.

This scenario is illustrated in the following figure:

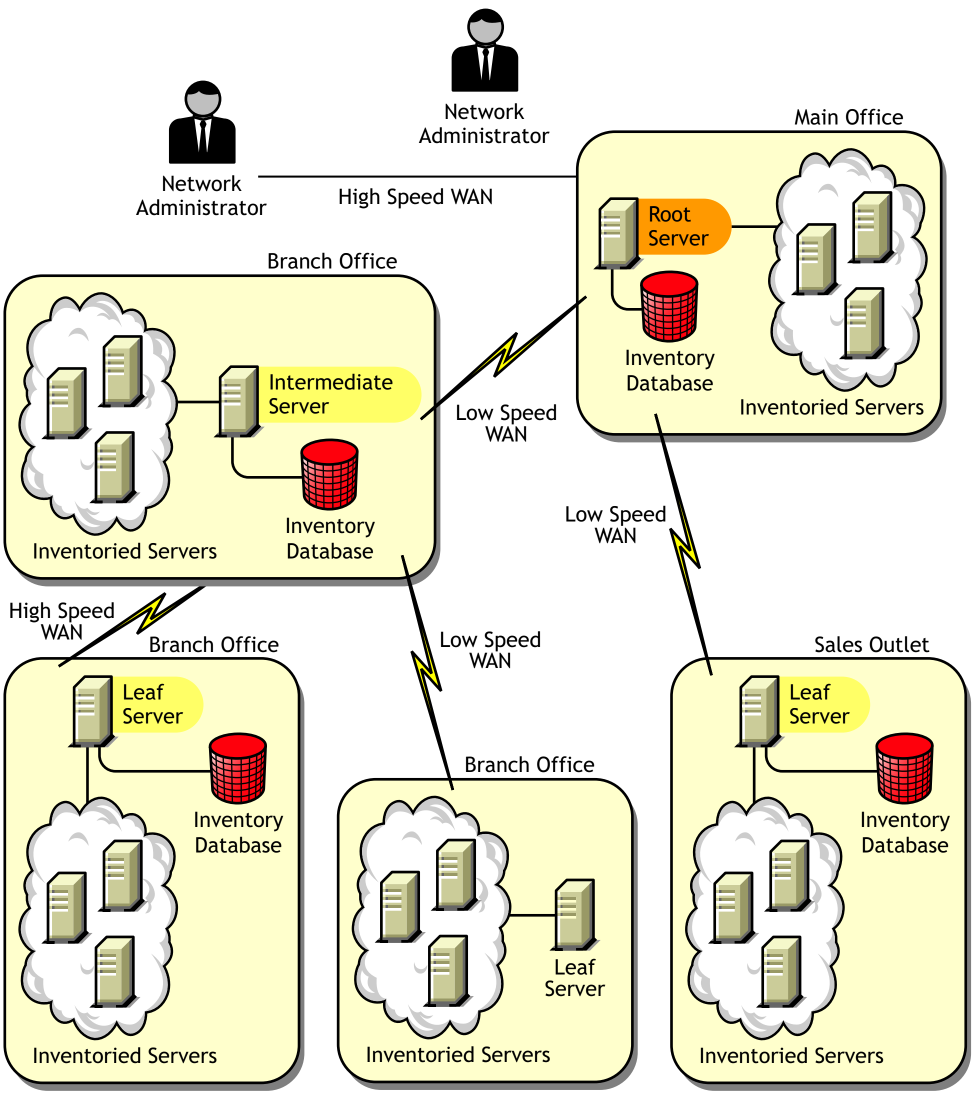


Scenario 4: Database on Inventory Servers and Intermediate Servers Connected to a Root Server

In this configuration, there are branch offices and a main office. Both branch offices store inventory information.

At one branch office, the Inventory server is a Leaf Server with Inventory Database, and the other branch office has a Leaf Server. At the next level, there is another branch office with an Intermediate Server with Database. The two branch offices at the lower level roll up data to this Intermediate Server. In turn, this Intermediate Server with Database rolls up data to the main office at the next level. There is also another sales outlet with a Leaf Server with Database at a sales outlet. This server directly rolls up data to the main office. The sales outlet and the two branch offices connect to the main office over low-speed WAN. One branch office connects to the main site over high-speed WAN.

This scenario is illustrated in the following figure:

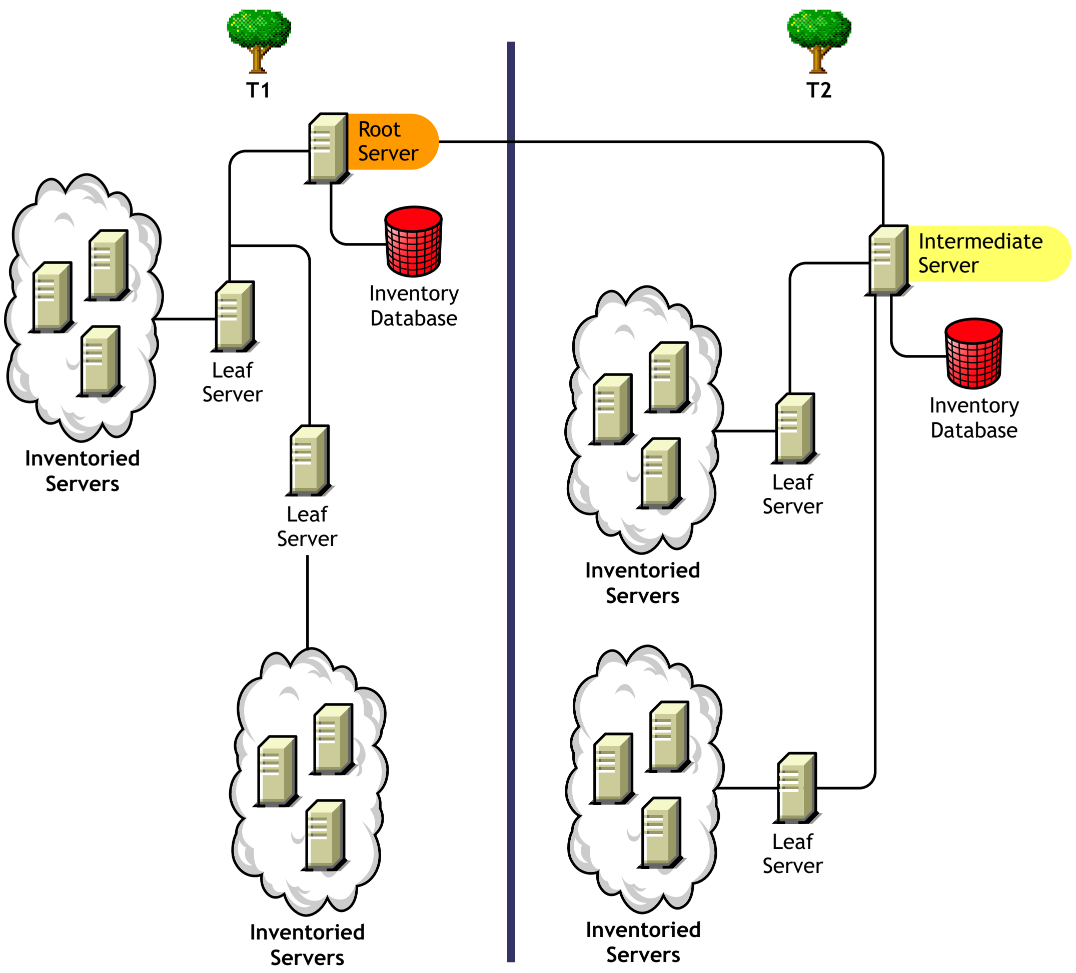


Scenario 5: Roll-Up of the Inventory information Across eDirectory Trees

In this configuration, you can deploy any of the previous scenarios. The highest-level Inventory server of one eDirectory tree rolls up the scan data 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 scan data is rolled up from the Intermediate Server to the Root Server residing on the eDirectory tree T1.

Scenario 6: 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_directory*\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:

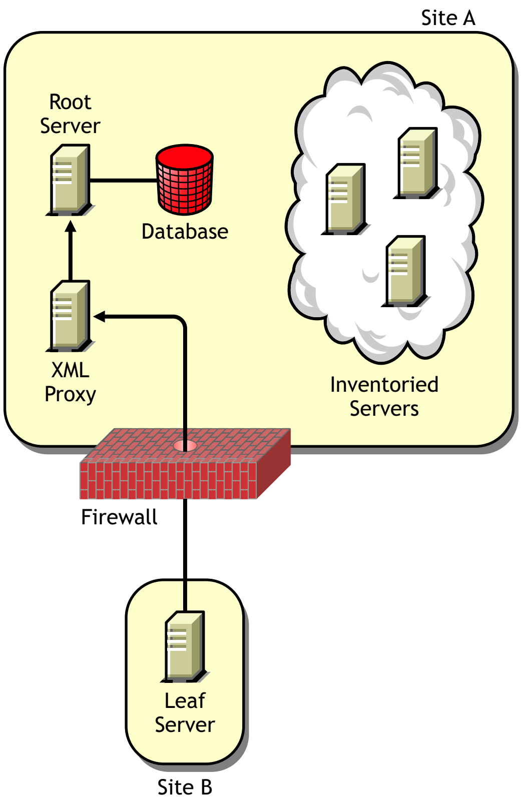
```
NDSTree=Target_eDirectory_tree_name
```

```
InventoryServiceDN=New_DN_of_the_Inventory_server
```

Scenario 7: 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:



Guidelines for Sending Inventory Information in a WAN

In this type of inventory deployment, the scanners transmit information to the servers over a WAN or dial-up connection.

- 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 server File System Service.
- You can attach inventoried servers to the server as determined by the number of connections supported by NetWare® or Windows* NT*/2000 servers up to a maximum of 5,000 inventoried servers.
- When you schedule the roll-up of data in the Inventory policies, we recommend the roll-up frequency should be at least one day. If the roll-up of scan data is scheduled too frequently, for example less than one hour, there may be some performance degradation of the inventory server.

Understanding the Effects of Server Inventory Installation

On the Inventory server, the ZfS 3 Server Inventory installation program does the following:

- On a NetWare Inventory server:
  - Copies the inventory related files to the *installation_directory*.
  - Copies the Server Inventory snap-in component to the ConsoleOne directory.
  - Creates an Inventory Service object (*servername*_ZenInvservice) 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 the scandir directory), and zeninvHostServer (DN of the server on which Inventory server is installed).
  - 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 (scandir) with the subdirectories (entpushdir, entmerge, and dbdir) in the specified volume on the Inventory server.
  - Creates the zenworks.properties file in sys:\system. This file contains the installation path of the Inventory server and the ZEN Web server.
  - Installs the ZEN 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 NT/2000 Inventory server:
  - Copies the inventory related files to the *installation_directory*.
  - Copies the Server Inventory snap-in component to the ConsoleOne directory.
  - Creates the scandir directory with the subdirectories.

    - Creates an Inventory Service object (*servername*_ZenInvservice) 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), and zeninvHostServer (DN of the server on which Inventory is installed).
    - The installation program assigns the Inventory Service object as trustee to itself.
    - On the Inventory server, the Inventory Service Manager is created as a service.
    - Edits the Registry settings to add the installation path of the Inventory server and the ZEN Web server.
    - On the Inventory server, the ZEN Web server 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 server, the Server Inventory installation program does the following:

- Installs the Sybase ASA 7.0.2.1583 (on NetWare) or Sybase ASA 7.0.2.1540 (on Windows NT/2000) and the Inventory database on the servers 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 NT/2000, the mgmtdb.db entries are added to the registry.
- Creates a Database object (*servername*_InventoryDatabase) for Sybase and configures the properties of the object.

On an existing ZENworks for Servers 2 installation, the installation program performs the following tasks in addition to the tasks performed for a fresh installation:

- On a NetWare Inventory server, the ZfS 3 Server Inventory installation deletes the following:
    - Inventory.ncf from the *installation_directory*\mwserver\bin directory
    - The ZfS 2 Inventory entries (gatherer.ncf, master.ncf, and storer.ncf) from sys:\system\autoexec.ncf.
    - The gpcsv and storer directories from *installation_directory*\mwserver.
- On the Database server, deletes zeninv.db from the list of databases that are loaded.

Setting Up the Inventory Database

The following sections contain detailed information to help you set up your Inventory database for Sybase and Oracle:

- "Setting Up the Inventory Database for Sybase" on page 685
- "Setting Up the Inventory Database for Oracle8i and Oracle9i" on page 690
- "Setting Up the Inventory Database for MS SQL Server 2000" on page 701

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 712.

Setting Up the Inventory Database for Sybase

This section contains the following information:

- “Manually Creating the Inventory Database Object for Sybase” on page 685
- “Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool)” on page 686
- “Understanding the Sybase Database Startup Parameters” on page 688
- “Optimizing the Performance of the Sybase Database” on page 688
- “Backing Up the Inventory Database Running Sybase” on page 689

Manually Creating the Inventory Database Object for Sybase

To manually create the Inventory database object 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.

- 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.

IMPORTANT: If the ZENworks database is located on a NetWare 4.x server, you must enter the server’s IP address in the Server IP Address or DNS Name field instead of adding the server’s object to the Server DN 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*

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 NT/2000 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 ZfS 3, 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 NT/2000 server, run the ntdbconfig.exe located in zenworks\dbengine directory. In the NTDBCONFIG dialog box, enter the new path of the 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 nt/2000, enter:

   ```
   mgmtdb3=c:\\zenworks\\inv\\db
   ```
5. Load the database, then enter **`mgmtdbs`** on NetWare servers, or on Windows NT/2000 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.
- **-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.
- ***database_installation_path*:** Specifies the installation path of the Inventory database. For example, c:\zenworks\inv\db\mgmtdb.db.

Optimizing the Performance of the Sybase Database

Increasing the database cache size improves database performance.

You can improve the performance of the Inventory database maintained in Sybase on NetWare or Windows NT/2000 Inventory servers. The default database cache size is 32 MB; however, this database cache size may not be adequate for large databases.

You should change the database cache size to an optimum size. You must also consider server memory size while assigning a cache size. For example, if you have 128 MB RAM, then a cache size of 32 MB is recommended.

To change the database cache size on the NetWare database server:

1. Close all connections to the Inventory database.
2. Quit the Sybase server.
3. Open the mgmtdbs.ncf file in the sys:\system directory.
4. Modify the -c parameter.

   For example, -c 64M sets the cache size to 64 MB.
5. Save the file.
6. On the server console, load the Inventory database. Enter **`MGMTDBS.`**

To change the database cache size on a Windows NT/2000 database server:

1. Stop the Sybase service.

   On Windows NT, in the Control Panel, double-click Services, select Novell Database - Sybase, then click Stop.

   On Windows 2000, in the Control Panel, double-click Administrative Tools, double-click Services, select Novell Database - Sybase, then click Stop.

2. On the database server, run the ntdbconfig.exe file from the dbengine directory.

   Ntdbconfig.exe is a ZENworks database configuration utility for the ZENworks database using Sybase on Windows NT/2000 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 688.

3. Modify the `-c` parameter.

4. Click OK.

5. Restart the Sybase service.

   On Windows NT, in the Control Panel, double-click Services, select Novell Database - Sybase, then click Start.

   On Windows 2000, in the Control Panel, double-click Administrative Tools, double-click Services, select Novell Database - Sybase, then click Start.

Backing Up the Inventory Database Running Sybase

ZfS provides an option to back up the Inventory database running Sybase from the ConsoleOne and Inventory database running Oracle 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.

To back up the database on NetWare or Windows NT/2000 servers:

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.

   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 just 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 just 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 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.

To restore the 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 NT, in the Control Panel, double-click Services, select Novell Database - Sybase, then click Stop.

   On Windows 2000, in the 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.

The backup tool creates a log file, backupst.txt, located in the consoleone*consoleone_version*\bin directory on NetWare and Windows NT/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.

Setting Up the Inventory Database for Oracle8i and Oracle9i

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

- "Setting Up the Inventory Database for Oracle8i" on page 690
- "Setting Up the Inventory Database for Oracle9i" on page 693
- "Manually Creating the Inventory Database Object for Oracle" on page 695
- "Loading the Inventory Database as a Separate Oracle Instance" on page 696
- "Optimizing the Performance of the Oracle Database" on page 698
- "Backing Up the Inventory Database Running Oracle" on page 699

Setting Up the Inventory Database for Oracle8i

- "Creating the Inventory Database for Oracle8i on a NetWare Server" on page 690
- "Creating the Inventory Database for Oracle8i on UNIX" on page 691
- "Creating the Inventory Database for Oracle8i on a Windows NT/2000 Server" on page 692

Creating the Inventory Database for Oracle8i on a NetWare Server

You must manually create the Inventory database for Oracle on NetWare servers.

Prerequisites for configuring the database include the following:

- ❑ Oracle8i (8.1.5.0.4) Enterprise Edition on NetWare must be installed on the server before configuring the Inventory database.
- ❑ To maintain the Inventory database in Oracle, Server Inventory requires that you have a minimum of twenty five Oracle user licenses.

- ❑ Oracle files should not be installed on an NFS-mounted volume on the file server.
- ❑ Oracle data files must reside on volumes that have block suballocation turned off.

Perform the following procedure to create the Inventory database on Oracle8i for NetWare:

1. Create a directory sys:\schema and copy all files from the following directories on the *ZENworks for Servers 3* product CD to the SCHEMA directory:
   - zfs\rminv\database\oracle\common
   - zfs\rminv\oracle\netwarespecific
2. Create the *user_specified_volumepath*\zenworks\inventory\oracle\database\trace directory structure. Here *user_specified_volumepath* refers to the user selected directory to create the database.
3. In sys:\schema_create.sql, replace all instances of **`oracle:`** with *user_specified_volumepath.*
4. In sys:\schema\init.ora, replace all instances of **`oracle:`** with *user_specified_volumepath*.
5. In sys:\schema_start.sql, replace all instances of **`oracle:`** with *user_specified_volumepath*.
6. Copy sys:\schema\init.ora to *user_specified_volumepath*\zenworks\inventory\oracle\database.
7. Copy sys:\schema_start.sql to *user_specified_volumepath*\zenworks.
8. At the command prompt, enter **`ORALOAD`** to start Oracle, if not started.
9. Ensure that no Oracle database is mounted.
10. At the command prompt, enter **`svrmgr31`** to load the Oracle Server Manager by
11. At the Oracle Server Manager prompt, enter **`@sys:\schema\schema.sql`**.

    Review the sys:\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 one or more of the following error messages: `Oracle not available, Out of space, Compilation error`.
12. At the Oracle Server Manager prompt, enter **`@<`*`volumepath`*`>\zenworks\_start.sql`** to start the Inventory database.

Creating the Inventory Database for Oracle8i on UNIX

Ensure that the following requirements are met:

- ❑ Oracle version

  On Linux* 6.0 or above: Oracle 8.1.5, 8.1.6 or 8.1.7 Enterprise Edition

  On Solaris* 6.2 or above on Sparc*/Intel*: Oracle 8.1.5, 8.1.6 or 8.1.7 Enterprise Edition
- ❑ System requirements

  Hard disk free space: 700 MB or above

  Primary memory: 512 MB or above
- ❑ To maintain the Inventory database in Oracle, Server Inventory requires that you have a minimum of twenty five Oracle user licenses.

You must manually create the Inventory database for Oracle8i on the UNIX* server by following the procedure below:

1. Log in as Oracle user.
2. Create a directory /schema and copy all files from the following directories on the *ZENworks for Servers 3* product CD to the schema directory:
   - zfs\rminv\database\oracle\common
   - zfs\rminv\oracle\unixspecific
3. Create the *user_specified_directory_path*/zenworks/inventory/oracle/database/trace directory structure.
4. In schema/init.ora, replace all instances of $HOME with *user_specified_directory_path*.
5. In schema/_start.sql, replace all instances of $HOME with *user_specified_directory_path*.
6. In schema/_create.sql, replace all instances of $HOME with *user_specified_directory_path*.
7. In schema/_schema.sql, replace all instances of $HOME with the schema directory created in Step 2.
8. Copy schema/init.ora to *user_specified_directory_path*/zenworks/inventory/oracle/database.
9. Copy schema/_start.sql to *user_specified_directory_path*/zenworks.
10. Ensure the Oracle services are up and running and no database is mounted.
11. At the command prompt, enter **`svrmgrl`** to load the Oracle Server Manager.
12. At the Oracle Server Manager prompt, enter **`@$HOME/schema/schema.sql`**

    Review the schema/inv.log file to ensure that the database has been created successfully. if the database has not been successfully created, schema/inv.log will contain the following error messages: `Oracle not available, Out of space, Compilation error`.
13. At the Oracle Server Manager prompt, enter **`@`*`user_specified_directory_path`*`/zenworks/_start.sql`** to start the Inventory database.

Creating the Inventory Database for Oracle8i on a Windows NT/2000 Server

You must manually create the Inventory database for Oracle on Windows NT/2000 servers.

Prerequisites for configuring the database include the following:

- ❑ Oracle 8.1.5, 8.1.6 or 8.1.7 Enterprise Edition must be installed on the server before configuring the Inventory database.
- ❑ To maintain the Inventory database in Oracle, Server Inventory requires that you have a minimum of twenty five Oracle user licenses.

Perform the following procedure to create the Inventory database on Oracle8i for Windows NT/2000:

1. Create a directory c:\schema and copy all files from the following directories on the *ZENworks for Servers 3* product CD to the schema directory:
   - zfs\rminv\database\oracle\common
   - zfs\rminv\oracle\winntspecific
2. Create the *user_specified_path*\zenworks\inventory\oracle\database\trace directory structure.
3. In c:\schema_create.sql, replace all instances of `d:` with *user_specified_path*.

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

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

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

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

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

9 Load the Oracle Server Manager by entering **`within a dos box: svrmgrl`**

10 At the Oracle Server Manager 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`

11 At the Oracle Server Manager prompt, enter **`@<path>\zenworks\_start.sql`** to start the Inventory database.

Setting Up the Inventory Database for Oracle9i

- "Creating the Inventory Database for Oracle9i on UNIX" on page 693
- "Creating the Inventory Database for Oracle9i on a Windows NT/2000/2003 Server" on page 694

Creating the Inventory Database for Oracle9i on UNIX

Ensure that the following requirements are met:

- ❑ Oracle9i release 2 must be installed on Linux or Solaris versions supported by Oracle9i
- ❑ System requirements

  Hard disk free space: 2 GB or above

  Primary memory: 512 MB or above
- ❑ To maintain the Inventory database in Oracle9i, Workstation Inventory requires that you have a minimum of 25 user licenses.
- ❑ ZENworks for Servers 3.0.2/SP 2 Interim Release 3 must be installed on the Inventory server. For more information about installing ZENworks for Servers 3.0.2/SP 2 Interim Release 3, see the *Readme of ZENworks for Servers 3.0.2/SP2 Interim Release 3* (http://support.novell.com).

You must manually create the Inventory database for Oracle9i on the UNIX server by following the procedure below:

1 Log in as an Oracle user.

2 Create a /schema directory and copy all files from the following directories to the schema directory:

- *zfs302_ir3.exe_extracted_directory*\zenworks\products\rminv\database\oracle9i\common
- *zfs302_ir3.exe_extracted_directory*\zenworks\products\rminv\database\oracle9i\unixspecific

3 Create the *user_specified_directory_path*/zenworks/inventory/oracle/database/trace directory structure.

4. In schema/init.ora, replace all instances of $HOME with *user_specified_directory_path*.
5. In schema/_start.sql, replace all instances of $HOME with *user_specified_directory_path*.
6. In schema/_create.sql, replace all instances of $HOME with *user_specified_directory_path*.
7. In schema/schema.sql, replace all instances of $HOME with the schema directory created in Step 2.
8. Copy schema/init.ora to *user_specified_directory_path*:/zenworks/inventory/oracle/database.
9. Copy schema/_start.sql to *user_specified_directory_path*/zenworks.
10. Ensure the Oracle services are up and running and no database is mounted.
11. At the command prompt, enter **`sqlplus /nolog`** to load the Oracle Server Manager.
12. At the Oracle Server Manager prompt (sqlplus prompt), enter **`@$HOME/schema/schema.sql`**.

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

Creating the Inventory Database for Oracle9i on a Windows NT/2000/2003 Server

You must manually create the Inventory database for Oracle9i on Windows servers.

Prerequisites for configuring the database include the following:

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

  Hard disk free space: 2 GB or above

  Primary memory: 512 MB or above
- ❑ ZENworks for Servers 3.0.2/SP 2 Interim Release 3 must be installed on the Inventory server. For more information about installing ZENworks for Servers 3.0.2/SP 2 Interim Release 3, see the *Readme of ZENworks for Servers 3.0.2/SP2 Interim Release 3* (http://support.novell.com).

To create the Inventory database on Oracle9i for Windows:

1. Create a directory c:\schema and copy all files from the following directories to the schema directory:
   - *zfs302_ir3.exe_extracted_directory*\zenworks\products\rminv\database\oracle9i\common
   - *zfs302_ir3.exe_extracted_directory*\zenworks\products\rminv\database\oracle9i\winntspecific
2. Create the directory structure: *user_specified_path*\zenworks\inventory\oracle\database\trace.
3. In c:\schema_create.sql, replace all instances of **d**: with *user_specified_path*.

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

5 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.

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

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

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

9 At the command prompt, enter **`sqlplus /nolog`** to load the Oracle server manager.

10 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`.

11 At the Oracle Server Manager prompt, enter **`@`*`user_specified_path`*`\zenworks\_start.sql`** to start the Inventory database.

Manually Creating the Inventory Database Object for Oracle

To manually create the Inventory database object for Oracle:

1 In ConsoleOne, right-click a location in the eDirectory tree for the database object, 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, 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 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.

- 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.

IMPORTANT: If the ZENworks database is located on a NetWare 4.x server, you must enter the server's IP address in the Server IP Address or DNS Name field instead of adding the server's object to the Server DN field.

3c Type the values for the following options:

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

- **Database (Write Only) Password:** *novell*

3d Click Apply.

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

3f Select Oracle, then click Default Settings.

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:** 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.

Loading the Inventory Database as a Separate Oracle Instance

The following sections explain the steps for configuring and running multiple Oracle database instances:

- "Configuring and Running Multiple Oracle Database Instances on a NetWare Server" on page 696
- "Configuring and Running Multiple Oracle Database Instances on a Windows NT/2000 Server" on page 698

Configuring and Running Multiple Oracle Database Instances on a NetWare Server

To configure and run multiple Oracle database instances:

1 Unload Oracle. At the database server prompt, enter **`oraunld`**.

2 Invoke the Net8 configuration utility. At the database server prompt, run `easycfg.ncf` to load the Net8 Easy configuration window.

3 Define a unique Oracle instance.

3a Click Config > Listener > Database > Add.

3b Assign values for Database Instance and Database Name in the Adding Instances Address window.

For example, assign Database Instance=*Indy* and Database Name=*mgmtdb*. In this configuration, the database instance is zfs. You can specify any database instance name. The Database Domain field should be left blank.

3c Click Accept, then click Save.

4 Configure the Listener for IPC. To run an Oracle system, the IPC and TCP addresses should be already be configured.

4a Click Config > Listener > Address. Ensure that IPC and TCP addresses are configured for the server.

The setting for IPC is *servername*_LSNR, and TCP is *IPaddress* or *hostname*. If these settings exist, click Cancel. Otherwise, assign the values for these settings > click Save.

5 Create an IPC alias.

5a Click Config > Database Alias. The window will list the aliases for IPC, SPX, TCP, and others. Click Add to add an alias name for the new instance.

Enter the following details:

- **Database Alias:** *servername-databaseinstance-IPC*. For example, the database alias is *austr*, where *austr* is the server name, and *indy* is the database instance created earlier.
- **Protocol:** *IPC*
- **Service/Host Name or Key Name:** *server_name_LSNR*
- **Database Instance:** *Indy*

5b Click Accept, then click Save.

5c To verify the configured alias name in the list window: Click Config > Database Alias, select the newly created alias, then click View.

View the properties of the database alias. Ensure that the properties are correct. If the property settings are incorrect, delete the alias (click Delete) and repeat Step 5.

6 Exit the EasyCfg tool. Click Config > Exit.

7 Create a password file for logging as *Internal* user for this instance. Enter **`load orapwd81 file=`*`oracle_volume`*`:`*`oracle_home`*`\`*`database`*`\pwd`*`database_instance`*`.ora password=`*`password`*`entries=2`** where *oracle_volume* is the NetWare volume name of your Oracle installation, *PWDdatabase_instance*.ORA is the password filename, and *password* is any password that you specify.

For example, `load orapwd81 file=oracle:\orahome1\database\pwdindy.ora password=`*`mgmtdb`* `entries=`*`2`*`.` This password file will be created in the *oracle_volume:*\DATABASE directory. Ensure that the file exists in the directory.

8 Load the Oracle NLM™ software. At the database server prompt, enter **`oraload`**.

9 To set the newly created ZfS instance, load the Oracle Server Manager. At the database server prompt, enter **`svrmgr31`**.

10 Enter the following commands: **`set instance servername-`*`databaseinstance`***. For example, `set instance `*`austr-indy-ipc`*`.` This displays that the newly created instance is started.

11 Enter **`connect internal/`*`password`*** where *password* is the password created in Step 7.

12 Mount the Inventory database.

13 Modify the _start.sql file located in *volumepath*\zenworks. Enter the following lines in the file:

```
set instance servername-databaseinstance-IPC
```

```
shutdown normal
```

14 Create the Database object. In ConsoleOne, right-click a location in the tree for the Database object, click New, click Object, click ZENworks Database, then click OK.

15 Type a name for the Database object, then click OK

16 Configure the Database server options of the Database object. For more information, see Step 3 on page 695 in "Manually Creating the Inventory Database Object for Oracle" on page 695

If you are loading multiple databases in separate Oracle instances, then each database reserves a separate Oracle SGA memory, where Oracle keeps all the database resources. In such environments, you should increase the amount of memory on the server. Refer to the documentation provided by Oracle.

Configuring and Running Multiple Oracle Database Instances on a Windows NT/2000 Server

To configure and run Oracle instances:

1 At the database server, run the Oracle Database Configuration Assistant. From the desktop Start menu, click Programs > Oracle > Database Administration > Oracle Database Configuration Assistant.

2 Click Create a Database > Next > Typical > Next > Copy Existing Database Files from the CD > Next.

3 Enter the following details:

- **Global Database Alias:** `mgmtdb.`*your_windows_NT/2000_name*
- **SID:** The value is automatically filled as *mgmtdb*.

4 Click Finish.

This allows for Oracle database creation. This process takes a significant amount of time. Ensure that the OracleServiceMGMTDB service is created and started.

5 Load the Inventory database.

Run the Oracle Server Manager. From the desktop menu, click Start > Run > SVRMGRL. Enter the following commands:

```
set instance mgmtdb
```

```
connect internal/password_for_administrator
```

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 INIT.ORA file in the ZENWORKS\DATABASE directory 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.

Oracle recommends that the database buffer cache for any Online Transaction Processing Application (OLTP) should have a hit ratio of about 90%, which is optimal.

The ZfS Inventory database on Oracle has an approximate 88% hit ratio with a database cache size of 24 MB for 128 MB RAM, which is about 20% of total memory.

If there is additional memory, you configure the database cache size by increasing the DB_BLOCK_BUFFERS parameter in the init.ora file.

Backing Up the Inventory Database Running Oracle

To back up the database running Oracle:

1 If the database server is up, stop the Storer service. At the database server console, enter **`StopSer Storer`**.

2 Load the Oracle Server Manager.

On NetWare server with Oracle, enter **`svrmgr31`**.

On Windows NT/2000 server with Oracle Enterprise Edition, from the taskbar, click Start > Run, enter **`svrmgrl`**.

3 Enter the following commands:

`set instance` ***`databaservername-databaseinstance-IPC`***, where *databaseinstance* refers to the database instance that you have set up earlier. See "Loading the Inventory Database as a Separate Oracle Instance" on page 696.

For example, **`set instance`** ***`austr-zfs3-ipc`***.

4 Connect as an administrator. For example, if the administrator's internal name is *internal*, at the Server Manager prompt, enter **`connect internal/`*****`password`***.

where *password* is the password created earlier. See "Loading the Inventory Database as a Separate Oracle Instance" on page 696.

4a At the Server Manager prompt, enter **`select name from v$datafile;`**

This displays the list of the data files that Server Inventory uses.

5 Ensure that no other databases are mounted. At the prompt, enter **`shutdown normal`**.

6 Disconnect and exit from the Server Manager. At the Server Manager prompt, enter **`disconnect;`**

Enter **`exit;`**

7 Copy the complete schema directory to a backup volume or disk.

After the backup is done, ensure that the backup copy of the database matches the original copy. Perform database verification to verify the integrity of the backup.

To verify the database integrity on a NetWare server with Oracle, enter **`load DBV81.NLM FILE=`***path_to_the_database_file* **`BLOCKSIZE=`*****`4096`***

To verify the database integrity on a Windows NT/2000 server with Oracle, enter **`DBV.EXE FILE=`*****`path_to_the_database_file`*** **`BLOCKSIZE=`*****`4096`***

Example: enter **`DBV.EXE FILE=`*****`c:\schema\database\cim1.ora`*** **`BLOCKSIZE=`*****`4096`***

Also, run this command for the following files: cim1.ora, cim2.ora, cim3.ora, cim4.ora, cim5.ora, cim6.ora, cim7.ora, cim8.ora, cim9.ora, cim10.ora, cim11.ora, sys1.ora, and ctl1.ora.

If the database backup is successful, ensure that there are no error messages on the verified pages. Ensure that the following displayed parameters display a zero value: TOTAL PAGES FAILING (DATA)=*0*, TOTAL PAGES FAILING (INDEX)=0, and TOTAL PAGES MARKED CORRUPT=0.

To restore the database:

1 If the Inventory database server is up, stop the Storer service. At the database server console, enter **`StopSer Storer`**.

2 Load the Oracle Server Manager.

On a NetWare server with Oracle, enter **`svrmgr31`**.

On a Windows NT/2000 server with Oracle Enterprise Edition, from the taskbar, click Start > Run, enter **`svrmgrl`**.

3 Connect as an administrator. For example, if the administrator's internal name is *internal*, at the Server Manager prompt, enter **`connect internal/`** ***`password_for_administrator`***.

4 Ensure that no other databases are mounted. Enter **`shutdown normal`**.

5 Disconnect and exit from the Server Manager. At the Server Manager prompt, enter **`disconnect;`**

Enter **`exit;`**

6 Copy the database from the backup location.

If you copy the database to a different location than the earlier location, modify the location in the following files to specify the new path:

- Edit the init.ora file located in \zfd3\oracle\database to specify the new path for the following parameters:

  ```
  control_files=location_of_ctll.ora\ctl1.ora
  ```

  ```
  background_dump_dest=location_of_trace_dir\trace
  ```

  ```
  user_dump_dest=location_of_trace_dir\trace
  ```

- Edit the _start.sql file in the sys:\system to specify the location of init.ora file in the following parameter:

  ```
  startup pfile=location_of_the_init.ora\init.ora
  ```

- Modify the location in the alterctrl.sql to specify new path.

  For example, modify the existing data:\zfd3\oracle\database path to oracle:\zfd3\oracle\database in alterctrl.sql.

  In this .SQL file, modify the path for the following parameters, if required.

  ```
  startup nomount pfile=database_path\INIT.ORA
  ```

  ```
  logfile group 1 'database_path\log1.ora' size 256K,
  ```

  ```
  logfile group 2 'database_path\log2.ora' size 256K
  ```

  ```
  datafile 'database_path\sys1.ora',
  ```

  ```
  'database_path\rbs1.ora',
  ```

```
'database_path\cim1.ora',

'database_path\cim2.ora',

'database_path\cim3.ora',

'database_path\cim4.ora',

'database_path\cim5.ora',

'database_path\cim6.ora',

'database_path\cim7.ora',

'database_path\cim8.ora',

'database_path\cim9.ora',

'database_path\cim10.ora',

'database_path\cim11.ora',

'database_path\tmp1.ora'
```

Save the changes.

7. Load the restored database.

Setting Up the Inventory Database for MS SQL Server 2000

This section provides information on the following topics:

- "Configuring the Inventory Database for MS SQL Server 2000" on page 701
- "Connecting the Inventory Server and ConsoleOne to the Inventory Database Running MS SQL 2000" on page 703

Configuring the Inventory Database for MS SQL Server 2000

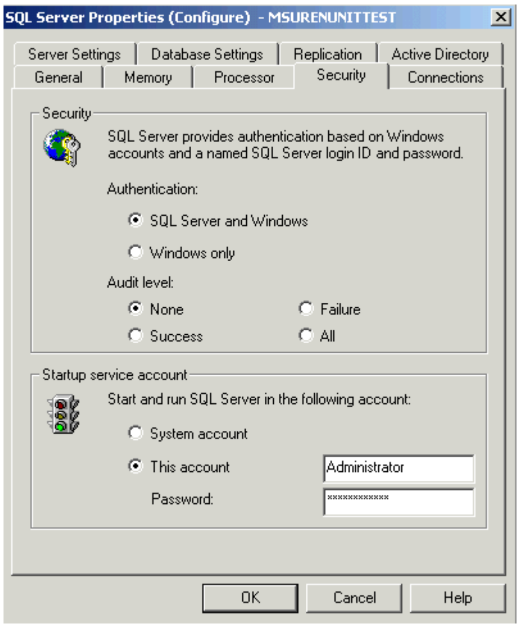
Prerequisites for configuring the database include the following:

- ❑ Microsoft SQL Server 2000 version 8.00.194 must be installed on the Windows NT/2000 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 data 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_for_servers_3_product_cd*\zenworks\products\rminv\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 > Microsoft SQL Server > 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.


SQL Server Properties (Configure) - MSURENUNITTEST

Security

SQL Server provides authentication based on Windows accounts and a named SQL Server login ID and password.

Authentication:
(●) SQL Server and Windows
( ) Windows only

Audit level:
(●) None ( ) Failure
( ) Success ( ) All

Startup service account

Start and run SQL Server in the following account:
( ) System account
(●) This account: Administrator
Password: ************

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:

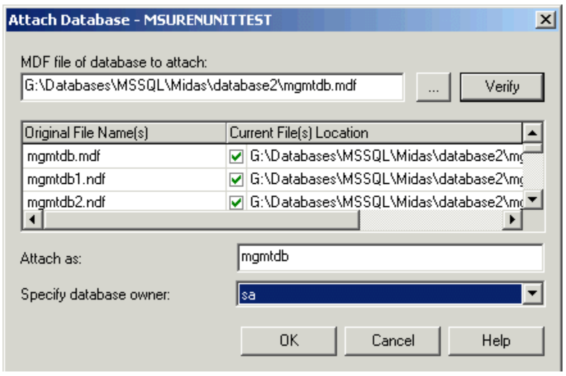
9a Click the Browse button to browse to and select mgmtdb.mdf as the .mdf database file to be attached.

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.


Attach Database - MSURENUNITTEST

MDF file of database to attach:
G:\Databases\MSSQL\Midas\database2\mgmtdb.mdf

Original File Name(s)	Current File(s) Location
mgmtdb.mdf	✓ G:\Databases\MSSQL\Midas\database2\mg
mgmtdb1.ndf	✓ G:\Databases\MSSQL\Midas\database2\mg
mgmtdb2.ndf	✓ G:\Databases\MSSQL\Midas\database2\mg

Attach as: mgmtdb

Specify database owner: sa

10 Select mgmtdb, then click the Tools menu, 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 > Open.

11c Select the createloginnames.sql query file from *zenworks_for_servers_3_product_cd*\zenworks\products\rminv\database\mssql directory.

11d Click Query > Execute.

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

```
New Login Created
```

12 Continue with "Connecting the Inventory Server and ConsoleOne to the Inventory Database Running MS SQL 2000" on page 703.

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

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/sql/downloads/2000/jdbc.asp).

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 all NetWare Inventory servers attached to the Inventory database mounted on MS SQL Server 2000, edit the sys:\system\invenv.ncf file to add the names of all the jar files of the JDBC driver in the following format:

```
envset tmppath=$tmppath;$root_dir\lib\msbase.jar

envset tmppath=$tmppath;$root_dir\lib\msutil.jar

envset tmppath=$tmppath;$root_dir\lib\mssqlserver.jar

...

...

envset tmppath=$tmppath;$root_dir\lib\jdbcdrv.zip
```

5 On all Windows NT/2000 Inventory servers attached to the Inventory database mounted on MS SQL Server 2000, do the following:

- Edit the *inventory_server_installation_directory*\wminv\bin\ zensetenv.ini file to append the following entry at the end of each line containing the classpath:

  ```
  ..\..\lib\msbase.jar;..\..\lib\msutil.jar;..\..\lib\mssqlserver.jar;
  ```

- Edit the *inventory_server_installation_directory*\wminv\bin\ invenv.bat file to add the following lines:

  ```
  set tmppath=%tmppath%;..\..\lib\msbase.jar
  ```

```
set tmppath=%tmppath%;..\..\lib\msutil.jar

set tmppath=%tmppath%;..\..\lib\mssqlserver.jar
```

6 On the machine running ZfS ConsoleOne with Inventory snap-ins, copy the msbase.jar, msutil.jar, and mssqlserver.jar files to the *consoleone_installation_directory*\lib\zen directory.

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

7a Right-click the container.

7b Click New, click Object, select ZENworks Database from the list of objects, then click OK.

7c Enter a name for the database object, then click OK.

8 Configure the Database server options of the Database object.

8a In ConsoleOne, right-click the database object, click Properties, then click the ZENworks Database tab.

8b 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.

- 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.

8c 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*

8d Click Apply.

8e To configure the JDBC Driver properties, click the JDBC Driver Information tab.

8f 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

8g Click Apply, then click Close.

Configuring Inventory Servers for Server Inventory

Based on the role on the Inventory server, you need to configure the settings of the Inventory server.

You can set policies to control how Inventory servers collect inventory. The Inventory policy settings configure the inventory scanning options for the selected Distributed Server Inventory Package. The Inventory policy settings stored in eDirectory are associated with an inventoried server object. Each inventoried server object has an associated Inventory policy package.

The following table lists the policies that you should configure after installing Server Inventory:

To set up this type of Inventory server:	Do this:
Standalone Server	1. Follow the steps in "Configuring the Server Inventory Policy" on page 707. 2. Follow the steps in "Configuring the Database Location Policy" on page 710.
Root Server	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Database Location Policy" on page 710.
Root Server with Inventoried Servers	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 707. 3. Follow the steps in "Configuring the Database Location Policy" on page 710.
Intermediate Server	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Roll-Up Policy" on page 711.
Intermediate Server with Database	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Roll-Up Policy" on page 711. 3. Follow the steps in "Configuring the Database Location Policy" on page 710.

To set up this type of Inventory server:	Do this:
Intermediate Server with Database and Inventoried Servers	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 707 3. Follow the steps in "Configuring the Roll-Up Policy" on page 711 4. Follow the steps in "Configuring the Database Location Policy" on page 710.
Leaf Server with Database	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 707. 3. Follow the steps in "Configuring the Roll-Up Policy" on page 711. 4. Follow the steps in "Configuring the Database Location Policy" on page 710.
Leaf Server	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 707. 3. Follow the steps in "Configuring the Roll-Up Policy" on page 711.
Intermediate Server with Inventoried Servers	1. Follow the steps in "Configuring the Inventory Service Object" on page 706. 2. Follow the steps in "Configuring the Server Inventory Policy" on page 707. 3. Follow the steps in "Configuring the Roll-Up Policy" on page 711.

IMPORTANT: After installing and configuring Server Inventory, you must run the Inventory Services on the Inventory server. For more information, see "Starting the Inventory Service" on page 712.

Configuring the Inventory Service Object

The Inventory Service object settings configure the scanning for the associated inventoried servers. From the Inventory Service Object property page, you can configure the following:

- Inventory Server Role
- Discard Scan Data Time
- Scan Directory Path
- Enable Scan

To open the Inventory Service Object properties page:

1 In ConsoleOne, right-click the Inventory Service object (*servername*_ZenInvservice), click Properties, then click the Inventory Service Object Properties tab.

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 661.

  Discard Scan Data Time: Any scan data files (.zip 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: Specify the volume or the directory of the Scan Directory (scandir) setting in the Inventory Service Object property page. The scandir directory path is the location on the Inventory server that stores the scan data files (.str files). The format of the Scan Directory Path is as follows: *Inventory_server_name\volume_of_the_server_directory*.

  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 scandir path.

  To modify the path: click the Browse button.

  - 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 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 712.

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 data will be 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:

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

2 Click Policies > General, NetWare, or Windows sub-options.

  To configure for both NetWare and Windows inventoried servers, click the General sub-option.

  NOTE: Do not select to configure policies in the Solaris or the Linux suboption as they are not supported.

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

4 Click Properties > the Server Inventory Policy tab.

5 Browse to select the DN of the Inventory Service object.

This setting specifies that the scanner will send the server scan data to this Inventory server.

NOTE: The Inventory Service object must be in the same eDirectory tree as the Server Inventory policy.

6 Select the DNS name or the IP address of the Inventory server.

7 If you want to send or roll-up the scan data to an Inventory server that is across the firewall, specify the IP address and the port number of the proxy server.

8 (Optional) To customize Inventory scanning, do the following:

8a Click the Hardware Scan tab.

This tab will be displayed while configuring the Server Inventory policy of the Windows servers.

8b Select the Enable DMI Scan option to include DMI scanning of inventoried Windows servers.

8c Select the Enable WMI Scan option to include WMI scanning of inventoried Windows servers.

8d Click the Software Scan tab.

8e Select the Enable the Software Scan option to enable software scanning of inventoried servers.

8f Click the Custom Scan Editor button to select the software that you want to scan for at the servers, then modify the list.

The Inventory scanner scans for the application information configured in the Custom Scan Editor.

For example, specify the following information in the Custom Scan Editor: Vendor Name=Microsoft; Product Name=Microsoft Office - Word; Product Version=10.0; FileName=winword.exe; File Size=10000.

In the Custom Scan Editor dialog box, the Inventory scanner scans for the winword.exe file having a size of 10000 bytes on the inventoried servers associated with the Server Inventory policy. If the file is found, the scanner stores "Microsoft;Microsoft Office - Word;10.0" for "winword.exe;10000" in the Inventory database.

The Custom Scan table lists the applications information that you want to scan for at the inventoried server such as vendor name, product name, product version, filename, and file size (in bytes).

You can enter a maximum of 1000 entries to the Custom Scan table.

Use the following methods to add the information of the applications to the Custom Scan table:

- Adding new entries

1. Click Add.
2. Specify the details.

You can specify details for one of the following attributes: Vendor Name, Product Name, or File\Location Name. But if you specify Product Version or File Size, you must also configure at least one of the following: Vendor Name, Product Name, or File\Location Name.

3. Click OK.

- Importing the contents from an existing file

1. Open a text editor and create a file with the following contents:
number_of_entries;number_of_columns
vendor_name;product_name;product_version;file_name;file_size_in_bytes;
...
vendor_name;product_name;product_version;file_name;file_size_in_bytes;

Fill in all the details of the application. The separator between the columns is a semicolon. Save the file as a text file with any extension you prefer. You can then import the file into the Custom Scan table.

A sample file is as follows:

```
2;5

Microsoft;Microsoft Office-Word;10.0;winword.exe;10000

Microsoft;Microsoft Office-Excel;5.0;excel.exe;50000
```

2. In the Custom Scan Editor dialog box, click Import.
3. Browse and select the file.
4. Click Open.

8g Select the Product Identification Number option to include scanning of product identification numbers of the Microsoft applications installed on the inventoried Windows servers.

8h Click the Configuration Editor tab.

8i Select the appropriate sub-option; Asset Information, Zipped Names, or SWRules.

8j Click Set Default to get the default settings.

8k If required, modify the settings of the configuration files, then click OK.

9 Click the Policy Schedule tab.

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

11 From the Distributed Server Package property page, click the Distribution tab, then click Add.

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

13 Click Apply, then click close.

14 In ConsoleOne, right-click the Inventory Service object (*servername*_ZenInvService), click Properties, then click the Inventory Service Object Properties tab.

15 Make sure the Enable Scan of Machines check box is selected, then click OK.

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

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, click Properties, then click Policies.
2. Select the check box under the Enabled column for the ZENworks Database policy.
3. Click Properties.
4. Click the Inventory Management tab.
5. Browse to the DN of the Inventory Database object, 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 NT/2000 server without eDirectory installed. To manually create the database object, see "Manually Creating the Inventory Database Object for Sybase" on page 685.

   For an Oracle database, you must create the database object and configure the object. For more information, see "Setting Up the Inventory Database for Oracle8i and Oracle9i" on page 690.
6. Click OK.
7. Click the Associations tab, then click Add.
8. Browse to select the container under which the Inventory Service object is located, then click OK.
9. Click Apply, then click Close.

To associate the Database object with an Inventory server:

1. In ConsoleOne, right-click the Server Package, click Properties, then click Policies.
2. Select the check box under the Enabled column for the ZENworks Database policy.
3. Click Properties.
4. Click the Inventory Management tab.
5. Browse to the DN of the Inventory Database object, 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 NT/2000 server without eDirectory installed. To manually create the database object, see "Manually Creating the Inventory Database Object for Sybase" on page 685.

   For an Oracle database, you must create the database object and configure the object. For more information, see "Setting Up the Inventory Database for Oracle8i and Oracle9i" on page 690.
6. Click OK.
7. Click the Associations tab, then click Add.

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

9 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 712.

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 scan data 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 Policy Packages container, click New > Policy Package > Server Package > RollupPolicy > Next.

2 Type a name for the Server Package, click Next, then click Finish.

3 In ConsoleOne, right-click the Server Package, click Properties, click Policies, then click General.

4 Check the check box under the Enabled column for the Rollup Policy.

5 Click Properties.

6 Click the Roll-up Policy tab, then click Roll-up Policy.

7 Browse to select the DN of the Inventory Service object, then click OK.

   Destination Server Object: You must specify the DN of the Inventory Service object at the next level Inventory server for moving the scan data 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 data to the same Inventory server. Also, you cannot specify the lower-level Inventory server as the next-destination server for roll-up of data.

8 Select the IP address or the DNS name of the next level Inventory server.

9 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.

10 Click the Associations tab, then click Add.

   The first time you enable the Roll-Up policy, you will prompted to associate the policy package. The policy you configured and enabled earlier will not be in effect until you associate this policy package with an Inventory server. Browse for the Inventory server that you want to associate the Roll-Up policy to, click OK > OK.

11 In ConsoleOne, right-click the Server Package, click Properties, then click Policies. Click NetWare or click Window sub-options.

12 Click the Roll-Up Policy row > Properties > Roll-Up Policy tab > Roll-Up Schedule. Modify the settings for scheduling the roll-up time, then click OK.

   When you schedule the roll-up of data in the Inventory policies, we recommend the roll-up frequency should be at least one day. If the roll-up of scan data is scheduled too frequently, for example less than one hour, there may be some performance degradation of the Inventory server.

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 712.

Starting and Stopping the Inventory Service

The section provides information on:

- "Starting the Inventory Service" on page 712
- "Stopping the Inventory Service" on page 713

Starting the Inventory Service

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

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

To manually start the Inventory services on the Windows NT Inventory server:

1. In the Control Panel, double-click Services.
2. Select Novell ZEN Inventory, then click Start.

To manually start the Inventory services on the Windows 2000 Inventory server:

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

To start a service on Windows NT/2000 server from the console prompt:

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.

After starting the Inventory service, make sure that the Inventory services are up and running.

To list all services:

- On a NetWare Inventory server, enter **`ListSer *`** at the console prompt.
- On a Windows NT/2000 Inventory server, enter **`ListSer "*"`** at the 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 844.

Stopping the Inventory Service

To stop the Inventory services on the 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.`**

To stop the Inventory services on the Windows NT Inventory server:

1. In the Control Panel, double-click Services.
2. Select Novell ZEN Inventory, then click Stop.

To stop the Inventory services on the Windows 2000 Inventory server:

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

To stop a service on Windows NT/2000 servers from the console prompt:

1. Go to the *Installation_directory*\inv\server\wminv\bin directory.
2. Enter **`StopSer`** ***`service_name`***.

   where *service_name* refers to an Inventory service.

Changing the Role of the Inventory Server

When you install ZfS, 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 (*servername*_ZenInvservice), click Properties, then click the Inventory Service Object Properties tab.
3. Choose the new role of the Inventory Service object, then click Apply.

   You will see a list of actions that you should follow based on the chosen role. For example, if you change the Root Server to a Root Server with Inventoried Servers, you need to configure the Server Inventory policy for the inventoried servers that you have attached. Similarly, to change the role to any other Inventory server, you need to follow the instructions to make the new role change effective. For more information, see "Configuring Inventory Servers for Server Inventory" on page 705.

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 NT/200 server, from the Services window, click Novell ZEN Inventory > Stop.

   To restart all Inventory Services:

   - At NetWare server console prompt, enter **`startinv`**
   - On the Windows NT/2000 server, from the Services window, click Novell ZEN Inventory > 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 715
- "Changing the Role of the Root Server with Inventoried Servers" on page 716
- "Changing the Role of the Intermediate Server" on page 717
- "Changing the Role of the Intermediate Server with Database" on page 718
- "Changing the Role of the Intermediate Server with Database and Inventoried Servers" on page 719
- "Changing the Role of the Intermediate Server with Inventoried Servers" on page 720
- "Changing the Role of the Leaf Server" on page 721
- "Changing the Role of the Leaf Server with Database" on page 722
- "Changing the Role of the Standalone Server" on page 722

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 data from this Inventory server.
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 data 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 data 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 data 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 data 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 data from this Inventory server.
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 data 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.

To change the role of the Standalone Server to ...	**Tasks:**
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 data 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 data 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 data 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 data 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 data 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 data from the Leaf Server with Database.

27 Understanding the Server Inventory Components

The following sections describe the Novell® ZENworks® for Servers (ZfS) 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 services:

Service Name	Description
Server Configuration Service	Loads the server configuration services
Inventory Scheduler Service	Loads the Inventory Scheduler
Selector Service	Loads the Selector

Service Name	Description
Receiver Service	Loads the Receiver
Sender Service	Loads the Sender
Storer Service	Loads the Storer
Scan Collector	Stores the .str files to the SCANDIR

Property File: There are property files that load the different services on the Inventory server depending on the role of the Inventory server. The name of the property file indicates the role of the Inventory server. Only the required services are loaded as per the role of the Inventory server. The property files should not be modified.

A sample role-based property file for a Leaf Server with Database is as follows:

```
[Server Configuration Service]

type = system

Load Sequence = 0

Load Option = auto

Class Name = com.novell.zenworks.desktop.inventory.
   servercommon.ServerConfig

Arguments =

[Upgrade Service]type = userLoad Sequence = 1Load Option =
   autoClass Name = com.novell.zenworks.desktop.inventory.
   upgradeService.UpgradeServiceArguments =

[Inventory Scheduler Service]

type = system

Load Sequence = 2

Load Option = auto

Class Name = com.novell.zenworks.desktop.inventory.
   servercommon.InventorScheduler

Arguments =

[Selector Service]

type = user

Load Sequence = 3

Load Option = auto

Class Name = com.novell.zenworks.desktop.inventory.
   selector.SelectorServiceInit

Arguments =

[Storer Service]

type = user

Load Sequence = 4
Load Option = auto

Class Name = com.novell.zenworks.desktop.inventory.
   storer.StorerServiceInit

Arguments =

[Sender Service]

type = user

Load Sequence = 5

Load Option = auto

Class Name = com.novell.zenworks.desktop.inventory.
   senderreceiver.control.SenderServiceInit

Arguments =

[NDSLookupForDB Service]

type = user

Load Sequence = 6

Load Option = manual

Class Name = com.novell.zenworks.desktop.inventory.
   dbutilities.NDSLookupForDB

Arguments = "WSDELETE.LOK"

[DBDelete Service]

type = user

Load Sequence = 7

Load Option = manual

Class Name = com.novell.zenworks.desktop.inventory.
   dbutilities.DBDelete

Arguments = "WSDELETE.LOK"

[DBBackup Service]

type = user

Load Sequence = 8

Load Option = manual

Class Name = com.novell.zenworks.desktop.inventory.
   dbutilities.DBBackup

Arguments = "Backup"
```

Do not modify these property files as services or the Service Manager cannot be loaded.

Depending on the role of the Inventory server, the server properties files include:

Server Type	Server Property File
Root Server	ROOT_DB.PROPERTIES
Root Server with Inventoried Servers	ROOT_DB_WKS.PROPERTIES
Intermediate Server	INT.PROPERTIES
Intermediate Server with Inventoried Servers	INT_WKS.PROPERTIES
Intermediate Server with Database	INT_DB.PROPERTIES
Intermediate Server with Database and Inventoried Servers	INT_DB_WKS.PROPERTIES
Leaf Server	LEAF_WKS.PROPERTIES
Leaf Server with Database	LEAF_DB_WKS.PROPERTIES
Standalone Server	STANDALONE.PROPERTIES

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

The contents of the config.properties file are as follows:

```
NDSTREE=treename

INVENTORYSERVICEDN=dn_of_the_inventory_service_object

SINGLETONPORT=65433

StoreRolledupAuditData=false

LDAPServerIP=LDAPserver_IPaddress

LDAPPort=LDAPserver_Portnumber
```

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.

The contents of the startinv.ncf file are as follows:

```
search add sys:\java\njclv2\bin

InvEnv

java -envDISPLAY=127.0.0.1:0 -sn"ZENworks Inventory Service"
   -noclassgc -DConfigFile=$inv_dir\properties\Config.
   properties -DDirectoryProp=$inv_dir\properties\
   Directory.properties -nsac -jszenWSInv -autounload
   -Xmx128m -classpath $tmppath;$classpath  com.novell.
   zenworks.desktop.inventory.servercommon.ZENWorksInven
   toryServiceManager
```

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.ZENWorksInventoryServiceManag
  er
  ```

- To start a 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 a 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 a 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 services, enter **`ListSer *`** at the Inventory server prompt.

- To stop the Inventory services on the Windows NT Inventory server:

  1. In the Control Panel, double-click Services.
  2. Select Novell ZEN Inventory, then click Stop.

- To stop the Inventory services on the Windows 2000 Inventory server:

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

Services on Windows NT/2000 Inventory Servers

On Windows* NT*/2000 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 a 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 a 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 services (ZENworks Inventory Service), use the Windows NT/2000 services from the desktop menu.

To list a 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

ZfS uses the following platform-dependent scanners to collect inventoried server hardware and software information:

- Invnatve.nlm, invalid.nlm and mpkscan.nlm to scan NetWare 5.1 and 6.0 inventoried servers.

  The NetWare scanner collects hardware details such as: hard disk drive, BIOS, processor, bus, display adapters, network adapter cards, sound cards, memory cards, serial ports, and parallel ports.

  The software scanning includes the following tasks:

  - Reading products.dat file and scanning for the Windows applications (.exe) that are installed locally on the inventoried servers.
  - Reporting the information about the scanned software like the vendor name, product name and the version number.
- Invnatve.nlm uses SNMP and NetWare API services to report inventory. Invnatve.nlm uses nwapi.map, smile.map and suppl.map configuration files for scanning various hardware inventory classes along with the source of information.

  For example, nwapi.map identifies the classes that are scanned using the NetWare API services, smile.map identifies the classes that are scanned using SNMP services and suppl.map identifies the classes that are scanned using System Management BIOS (SMBIOS) services. The list of the configuration files used by invnatve.nlm are identified in the hwinvsrc.ini file.
- Invalid.nlm reads the System Management BIOS structures and scans for certain hardware inventory such as BIOS, processors, serial and parallel ports, cache, sound adapters and display adapters.

- Mpkscan.nlm scans for the software inventory on the NetWare inventoried servers.

  NOTE: The NetWare scanner that ships with ZfS 3 does not report inventory of processors for multi-processor (MPK) servers if it is not available in the SMBIOS. To scan inventory on a multiprocessor (MPK) NetWare server, you must refer to the TID 2961928 in the Knowledgebase at Novell Technical Services (http://support.novell.com) and install the latest patch.

  The NetWare scanner also does not report the virtual memory size.

- Invnatve.dll and invscan.exe to scan Windows NT/2000 inventoried servers.

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

  NOTE: On Windows NT/2000 servers, the ZfS 3 Inventory scanner will not scan for Tape drives, Jaz* drivers, and Zip* drives. The Inventory scanner will not accurately report the physical memory if the memory is greater than 2 GB.

  Invscan.exe leverages DMI and WMI support on the inventoried server to report hardware inventory.

The scan information collected by the scanners is stored as scan data files (.str) locally on the inventoried server. The scanned data is transferred to the Inventory server using the XML-RPC protocol. The Scan Collector receives and stores the .str files in the scan directory (scandir).

The following sections contain detailed information about the Inventory scanners:

- "How the Scanners Collect server Inventory Data" on page 731
- "Scanning Process Flowchart" on page 733
- "Software Information Collected by the Scanners" on page 733
- "DMI-Compliant Scanners" on page 734
- "WMI-Compliant Scanners" on page 735
- "SNMP-Compliant Scanners" on page 736
- "Hardware Data Collected by the Scanners" on page 736

How the Scanners Collect server Inventory Data

The scanning process is as follows:

- Subscribers must be installed and configured on the inventoried servers for Tiered Electronic Distributions (TED).
- The Server Inventory policy lets you configure the scanning schedule based on which the policy engine schedules and enforces scanning at the inventoried server.
- The Policy Enforcer triggers the Scanner, which reads the following inventory settings defined in the Inventory Service object and the Server Inventory policy.
  - **Enable Scan of Machines:** By default, the Enable Scan of Machines option is selected in the Inventory Service object property page. The Scanner collects the inventory information of the inventoried servers.
  - **Scan Directory Path:** Browse to select the DN of the Inventory Service object of the destination Inventory server.

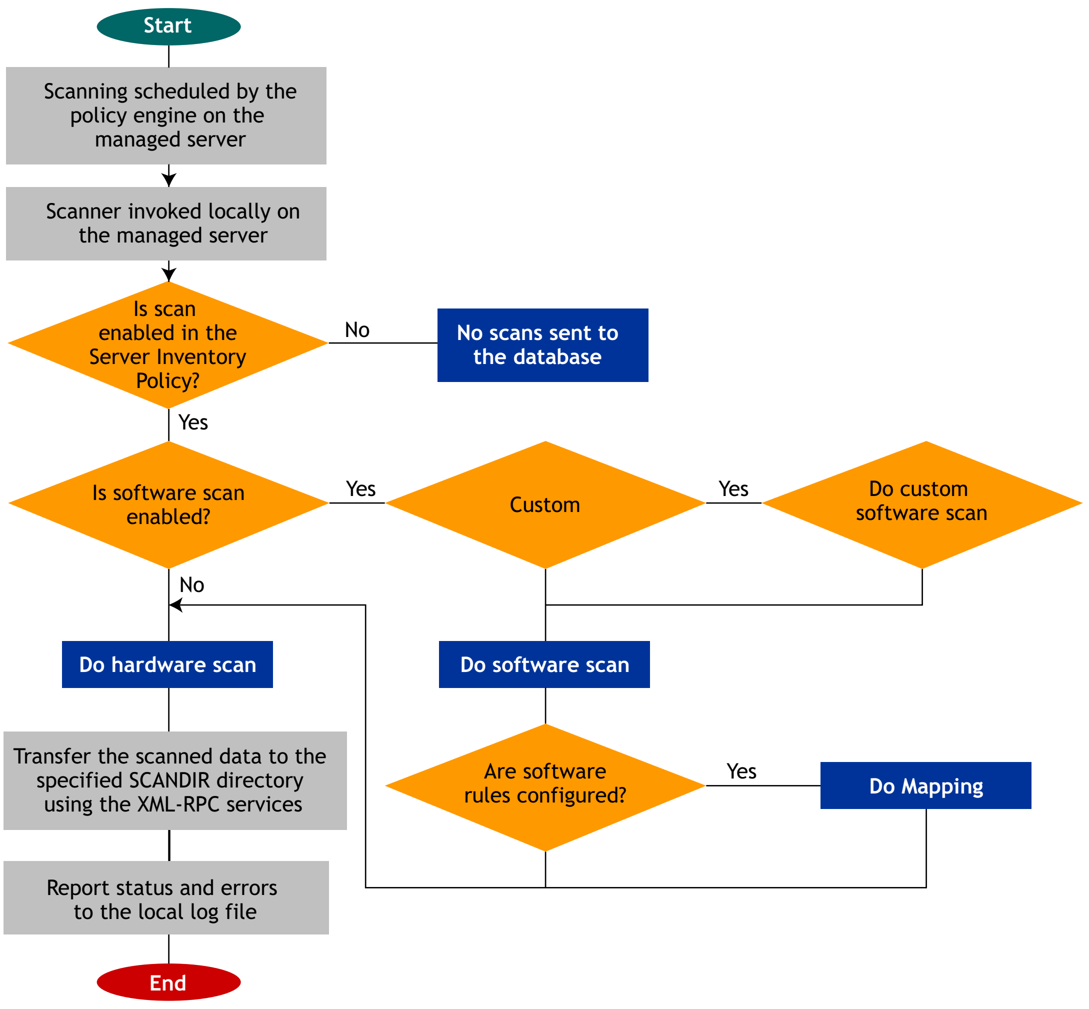
- **Enable Software Scan:** If the Enable Software Scan option is enabled in the Server Inventory policy, the Scanner collects information about software applications.
- **Custom Scan Editor:** If the Enable Software Scan option is enabled, the Scanner reports the software information. You configure the applications that you want the Scanner to collect information by using the Custom Scan Editor. You can configure the Custom Scan Editor to find information about the products that are installed on the inventoried server and do not have any information in their respective header files or the information in the header files is not accurate. The Inventory scanner picks the information configured in the Custom Scan Editor and stores it into the database
- **Configuration Editor:** If the Enable Software Scan option is enabled, you can configure the software rules to be referred while reporting the software inventory. For Windows inventoried servers, you can additionally configure for the asset information and the zip names that are to be referred during hardware inventory.
- **Enable DMI Scan:** If the server is instrumented for DMI, the scanners query the DMI Service Layer. For more information, see "DMI-Compliant Scanners" on page 734
- **Enable WMI Scan:** If the servers are WMI-compliant, the scanners also collect the hardware data by querying the WMI information. The scanners also probe the servers for hardware data. For more information, see"DMI-Compliant Scanners" on page 734

  We recommend that you instrument DMI/WMI on your inventoried servers and install DMI/WMI components that are supplied by the vendors.

- The scan data of each inventoried server is stored as .str files in the scandir directory on the Inventory server. The .str file follows the filename convention: `macaddress_gmt_sequencenumber.str`, where *macaddress* is the MAC Address of the inventoried server, *gmt* is the time at which the inventoried server is scanned for the first time, and *sequencenumber* is the internal sequencing number of the inventoried server, which is always zero ('0') to indicate a full scan. For example, 00508b12b2c4_944029836000_0.str is the .str file for the server with the MAC address of 00508b12b2c4, the GMT of 944029836000, and the internal sequencing number of 0.
- The Scanner reports errors in local log files only (invagent.log and invnatve.log). The log file is stored in the sys:\etc directory on NetWare inventoried servers and in the windows directory or the temp directory on Windows NT/2000 inventoried servers.

Scanning Process Flowchart

The following flowchart illustrates the hardware and software scanning process:


Start
Scanning scheduled by the policy engine on the managed server
Scanner invoked locally on the managed server
Is scan enabled in the Server Inventory Policy?
No
No scans sent to the database
Yes
Is software scan enabled?
Yes
Custom
Yes
Do custom software scan
No
Do hardware scan
Do software scan
Transfer the scanned data to the specified SCANDIR directory using the XML-RPC services
Are software rules configured?
Yes
Do Mapping
Report status and errors to the local log file
End

Software Information Collected by the Scanners

The scanners follow this process for software scanning:

- They collect the information about the software on the inventoried servers.
- They customize the software scanning using the Custom Scan Editor.

  By default, the software scanning includes collecting version information of files with .exe file extensions.

  If Windows software applications (files with .exe extensions) are installed on the NetWare inventoried server, the scanner interprets the Microsoft Portable Executable (PE) format of these files to collect and report software information.

  If the software applications are installed using Microsoft* Installer on the Windows NT/2000 inventoried server, the scanners use the information from Microsoft Installer (MSI). Otherwise, the scanners collect the software information from the header of the software application files.

- They report the information about the scanned software, such as name of the software product for each product version and the software vendor.

After the scan data is stored in the database, you can view or query the software information.

If you want to know the mode used by the Scanner to scan the inventoried server, see the inventory summary of the inventoried server > Hardware/Software Inventory > Software > Inventory Scanner Information > Scan Mode.

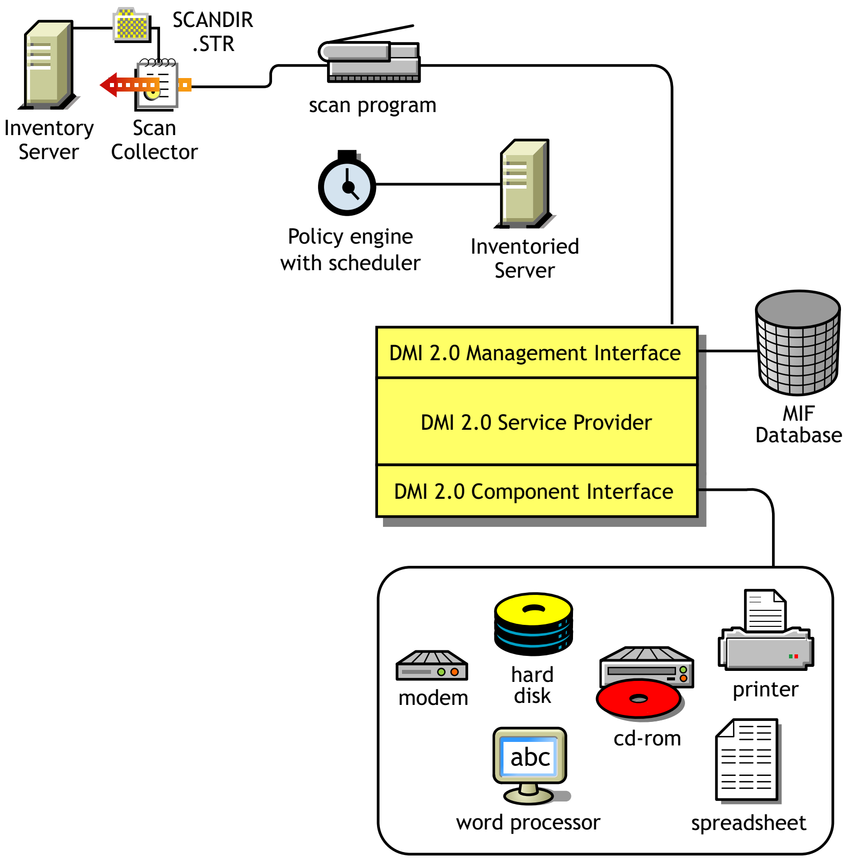
If you want to know the technology available on the inventoried server such as DMI, WMI, and others, see the inventory summary of the inventoried server > Hardware/Software Inventory > General > System Information > Management Technology.

DMI-Compliant Scanners

The scanners for scanning inventoried servers (Windows NT/2000) 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).

To scan the DMI data of the inventoried servers, you need to instrument the inventoried server by installing the vendor-specific components.

For example, if you have a Compaq* Family model server running under Windows NT, download the Management Product software - Compaq Insight Management Desktop Agents software for Windows NT from the Compaq web site.

For Dell* servers, access the DM/Desktop Management Utilities software from the Dell web site.

WMI-Compliant Scanners

The scanners collect hardware data from Windows NT versions 4.0/2000 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 the Microsoft WMI Web site (http://www.microsoft.com/hwdev/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 NT versions 4.0 /2000 inventoried servers only.

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

To obtain WMI information from the Windows NT version 4.0 inventoried server:

1 Download Microsoft's Windows Management Instrumentation - Core Software Installation from the Microsoft WMI Web site (http://msdn.microsoft.com/downloads/default.asp?url=/downloads/sample.asp?url=/msdn-files/027/001/576/msdncompositedoc.xml).

   Only the WMI Core Software Installation download is required to instrument an inventoried server for WMI. To troubleshoot any WMI related problems, you can avail WMI SDK download.

   IMPORTANT: On Windows 2000 servers, WMI Core Software is already installed.

2 Install WMI Core Software on Windows NT version 4.0 servers.

SNMP-Compliant Scanners

The Windows scanners use the Simple Network Management Protocol (SNMP) to report network adapter information and network information such as IP address, subnet mask, etc.

The NetWare scanners collect certain hardware (device) and network information based on SNMP. Additionally, the NetWare scanner also uses SNMP to report software installed and registered in products.dat. The NetWare scanner uses SNMP v2.0 and the services of hostmib.nlm, ipxrtr.nlm, and ipxrtrnm.nlm. For more information on the details about dependent MIBs and etc., see "Hardware Data Collected by the Scanners" on page 736

Hardware Data Collected by the Scanners

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

Scan Data	SNMP Details	SMBIOS Details
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
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

Scan Data	SNMP Details	SMBIOS Details
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
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

Scan Data	SNMP Details	SMBIOS Details
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
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

Scan Data	SNMP Details	SMBIOS Details
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
LogicalDrive.DeviceID	Not applicable	Not applicable
LogicalDrive.VolumeSerialNumber	Not applicable	Not applicable

Scan Data	SNMP Details	SMBIOS Details
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

The Inventory Agent collect the following hardware information on the Windows NT/2000 inventoried servers:

Scan Data	DMI Class and Attribute	WMI Class and Attribute
System.Type	Not applicable	Win32_SystemEnclosure.Manufacturer
System.MachineName	Not applicable	Not applicable
System.AssetTag	DMTF|System Enclosure|001.2	Win32_SystemEnclosure.SMBIOSAssetTag
System.Model	Not applicable	Win32_SystemEnclosure.Model
System.ModelNumber	Not applicable	Win32_SystemEnclosure.SerialNumber
System.SystemIdentifier(GUID)	Not applicable	Not applicable
System.ManagementTechnology	Not applicable	Not applicable
System.DNName	Not applicable	Not applicable
System.TreeName	Not applicable	Not applicable
NetworkAdpater.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MACAddress (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
IP.Address	Not applicable	Win32_NetworkAdapterConfiguration.IPAddress (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
IP.Subnet (Subnet Mask)	Not applicable	Win32_NetworkAdapterConfiguration IPSubnet (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
NetworkAdapter.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MACAddress (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
IPX.Adress	Not applicable	Win32_NetworkAdapterConfiguration.IPXAddress (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
NetworkAdapter.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MACAddress (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
DNS.HostName	Not applicable	Win32_NetworkAdapterConfiguration.DNSHostName + DNSDomain (Only on Windows NT/2000, get through association Win32_NetworkAdapterSetting)
Modem.Description	Not applicable	Win32_POTSModem.Description
Modem.Name	Not applicable	Win32_POTSModem.Name

Scan Data	DMI Class and Attribute	WMI Class and Attribute
Modem.Vendor	Not applicable	Win32_POTSModem.ProviderName
Modem.DeviceID	Not applicable	Win32_POSTSModem.DeviceID
NetworkAdapter.Speed	DMTF|Network Adapter 802 Port|001.5	Win32_NetworkAdapter.MaxSpeed (Only on Windows NT, when Win32_NetworkAdapter.AdapterType=Ethernet 802.3 or Fiber Distributed Data Interface (FDDI))
NetworkAdapter.Name	Not applicable	Win32_NetworkAdapter.Name (Only on Windows NT, when Win32_NetworkAdapter.AdapterType=Ethernet 802.3 or FDDI)
NetworkAdapter.PermAddress	DMTF|Network Adapter 802 Port|001.2	Win32_NetworkAdapter.PermanentAddress (Only on Windows NT, when Win32_NetworkAdapter.AdapterType=Ethernet 802.3 or FDDI)
NetworkAdapter.AdapterType	Not applicable	Win32_NetworkAdapter.AdapterType (Only on Windows NT, when Win32_NetworkAdapter.AdapterType=Ethernet 802.3 or FDDI)
NetworkAdapter.ProviderName	Not applicable	Win32_NetworkAdapter.Manufacture (Only on Windows NT, when Win32_NetworkAdapter.AdapterType=Ethernet 802.3 or FDDI)
NetworkAdapter.DriverDescription	DMTF|Network Adapter Driver |001.Driver Software Description	Win32_SystemDriver.Description (Only on Windows NT, when Win32_SystemDriver.Name=Win32_NetworkAdapter .ServiceName)
NetworkAdapter.DriverName	DMTF|Network Adapter Driver |001.Driver Software Name	Win32_SystemDriver.PathName (Only on Windows NT, when Win32_SystemDriver.Name=Win32_NetworkAdapter .ServiceName)
NetworkAdapter.DriverVersion	DMTF|Network Adapter Driver |001.Driver Software Version	Not applicable
Login.CurrentLoggedinUser	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

Scan Data	DMI Class and Attribute	WMI Class and Attribute
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	Not applicable	CIM_Processor.Description
Processor.Name	Not applicable	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
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.TiggerType	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.Keybo ard.Type	CIM_Keyboard.Description
Keyboard.Fkeys	Not applicable	CIM_Keyboard.NumberOfFunctionKeys
Keyboard.Delay	Not applicable	Not applicable
Keyboard.TypematicRate	Not applicable	Not applicable

Scan Data	DMI Class and Attribute	WMI Class and Attribute
Monitor.NumberOfColorPlanes	Not applicable	Win32_VideoController.NumberOfColorPanes
Monitor.HorizontalResolution	DMTF|Video|004.Current Horizontal Resolution	Win32_VideoController.CurrentHorizontalResolution
Monitor.VerticalResolution	DMTF|Video|004.Current Vertical Resolution	Win32_VideoController.CurrentVerticalResolution
Monitor.DisplayType	DMTF|Video|004.Video Type	Win32_VideoController.VideoArchitecture
Monitor.MemoryType	DMTF|Video|004.Video Memory Type	Win32_VideoController.VideoMemoryType
Monitor.MaxMemorySupported	DMTF|Video|004.Video RAM Memory Size	Win32_VideoController.MaxMemorySupported
Monitor.Bitsperpixel	DMTF|Video|004.Current Number of Bits per Pixel	Win32_VideoController.CurrentBitsPerPixel
Monitor.ControllerDescription	DMTF|Video|004.Video Controller Description	Win32_VideoController.Description
Monitor.MaxRefreshrate	DMTF|Video|004.Maximum Refresh Rate	Win32_VideoController.MaxRefreshRate
Monitor.MinRefreshrate	DMTF|Video|004.Minimum Refresh Rate	Win32_VideoController.MinRefreshRate
Mointor.DACType	Not applicable	Win32_VideoController.AdapterDACType
Monitor.ChipSet	Not applicable	Not applicable
Monitor.ProviderName	Not applicable	Not applicable
Monitor.VideoBIOSManufacturer	DMTF|Video BIOS|001.BIOS Manufacturer	CIM_VideoBIOSElement.Manufacturer
Monitor.VideoBIOSVersion	DMTF|Video BIOS|001.Video.BIOS Version	CIM_VideoBIOSElement.Version
Monitor.VideoBIOSReleaseDate	DMTF|Video BIOS|001.Video.BIOS Release Date	CIM_VideoBIOSElement.InstallDate
Monitor.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

Scan Data	DMI Class and Attribute	WMI Class and Attribute
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
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	DMTF|Disks|003.Interface Description (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	DMTF|Disks|003.Number of Physical Cylinders	Not applicable
FloppyDrive.NumberOfHeads	DMTF|Disks|003.Number of Physical Heads	Not applicable
FloppyDrive.SectorsPerTrack	DMTF|Disks|003.Number of Physical Sectors Per Track	Not applicable
FloppyDrive.Size	DMTF|Disks|003.Total Physical Size	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

Scan Data	DMI Class and Attribute	WMI Class and Attribute
CDROMDrive.Caption	Hard code: Cdrom Device (when DMTF|Disks|001.Logical Drive Type = 6)	Win32_CDROMDrive.Caption
HardDrive.Media Type	DMTF|Disks|003.Removabl e 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
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))
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))
OperatingSystem.OSType	Not applicable	Win32_OperatingSystem.OSType
OperatingSystem.Version	Not applicable	Win32_OperatingSystem.Version

Scan Data	DMI Class and Attribute	WMI Class and Attribute
OperatingSystem.Codepage	Not applicable	Win32_OperatingSystem.CodeSet
OperatingSystem.InstallDate	Not applicable	Win32_OperatingSystem.InstallDate
OperatingSystem.TotalSwapSpace Size	Not applicable	Not applicable
OperatingSystem.Description	Not applicable	Win32_OperatingSystem.Caption
OperatingSystem.OtherTypeDescrip tion	Not applicable	Win32_OperatingSystem.OtherTypeDescription
OperatingSystem.VirtualMemorySize	DMTF|System Memory Settings|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
InventoryScanner.ScanMode	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.ErrorMethodology
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

Scan Data	DMI Class and Attribute	WMI Class and Attribute
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
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

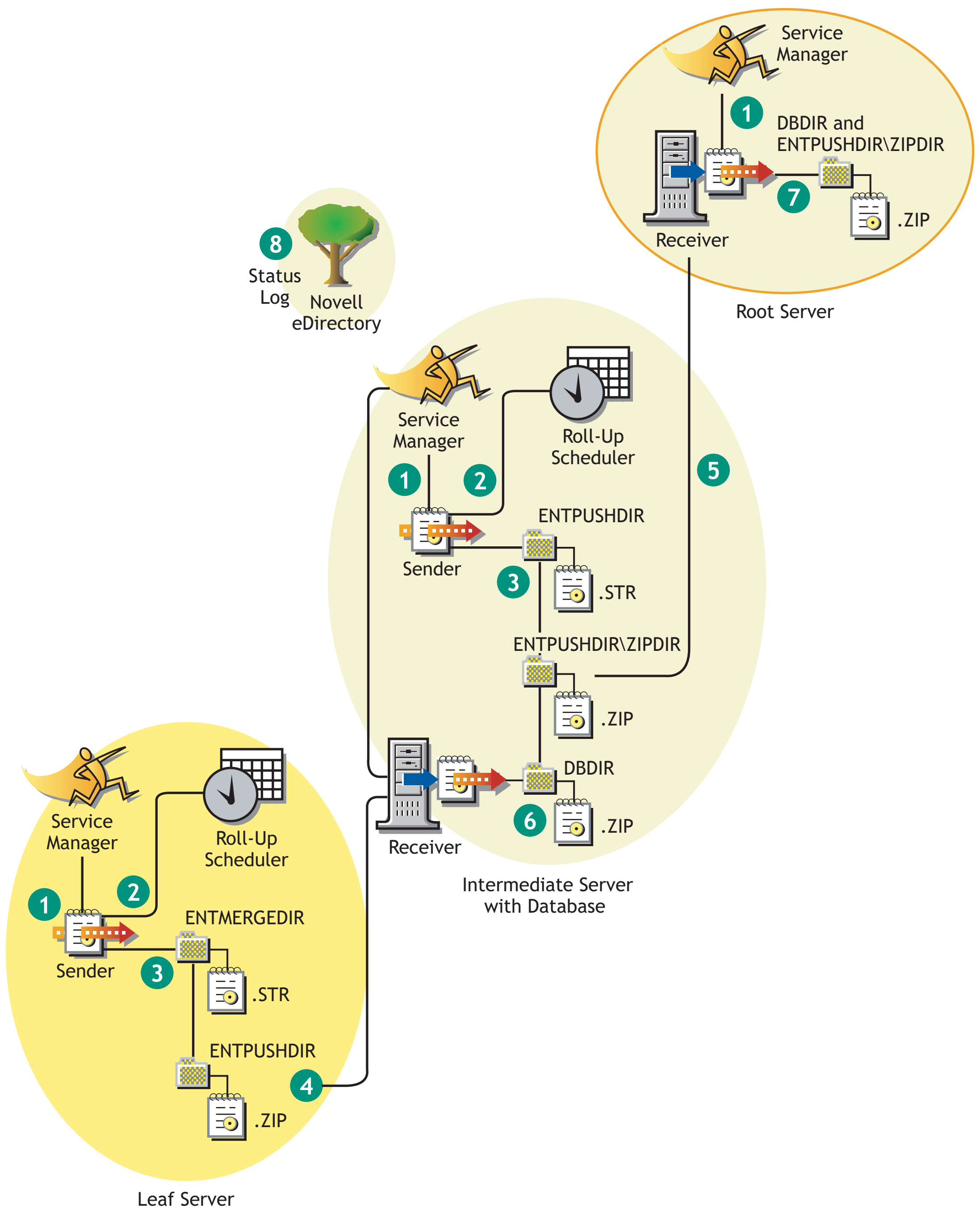
Scan Data	DMI Class and Attribute	WMI Class and Attribute
PowerSupply.Power	DMTF|Power Supply|002.Total Output Power	CIM_UninterruptiblePowerSupply.TotalOutputPower
DMA.Number	DMTF|DMA|001.DMA Number	CIM_DMA.DMAChannel
DMA.Description	DMTF|DMA|001.DMA Description	CIM_DMA.Description
DMA.Availability	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
UCS.PrimaryOwnerContact	DMTF|General Information|001.3	CIM_UnitaryComputerSystem.PrimaryOwnerContact
UCS.PrimaryOwnerName	DMTF|General Information|001.4	CIM_UnitaryComputerSystem.PrimaryOwnerName
Pointing Device.DeviceType	Not applicable	CIM_PointingDevice.PointingType
Pointing Device.Type	DMTF|Pointing Device|Pointing Device Interface	CIM_PointingDevice.Name
Pointing Device.NumberOfButtons	DMTF|Pointing Device|Number Of Buttons	CIM_PointingDevice.NumberOfButtons
Pointing Device.DriverName	DMTF|Pointing Device|Pointing Device Driver Name	CIM_PointingDevice.Name
Pointing Device.DriverVersion	DMTF|Pointing Device|Pointing Device Driver Version	Not applicable
Pointing Device.IRQ	DMTF|Pointing Device|Pointing Device IRQ	Not applicable
Slot.Description	DMTF|System Slots|003.Description	Not applicable
Slot.MaxDataWidth	DMTF|System Slots|003.MaxDataWidth	Not applicable
Slot.ThermalRating	DMTF|System Slots|003.Slot Thermal Rating	Not applicable

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 Scan Collector uses the ZEN Web Server to process the XML-RPC requests. The following sections contain more information:

- "Understanding the Sender" on page 753
- "Understanding the Receiver" on page 754
- "Understanding the Compressed Scan Data File" on page 754
- "Sender-Receiver Directories" on page 755

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.
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 760 for a quick reference table of Inventory server components.

The flow of information from the Sender in the roll-up of scan data 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. For more information, see "Understanding the Compressed Scan Data File" on page 754.
2. The Sender creates a new record in the zeninvRollUpLog attribute of the Inventory Service object (ZenInvservice) 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 data, 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 scan data 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 843.

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 data 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 760 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 843.

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 1,000 .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 scan data 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 760 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 843.

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 760 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 scan data 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 843.

Understanding the Inventory Removal Service

The Inventory Removal service is a manual service that runs on the Inventory server. You can remove the unwanted, redundant, or obsolete inventoried servers from the Inventory database using the Inventory Removal service. The Inventory Removal service removes the inventoried servers from the Inventory database through using the inventoryremovallist.txt file. To understand the Inventory Removal Service synchronization, see "Using the Inventory Removal Service for Synchronization" on page 758.

The inventoryremovallist.txt file contains a list of inventoried servers that have to 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 or name of the inventoried server (as stored in the Inventory database)
to be removed from the Inventory database
DN or name of the inventoried server (as stored in the Inventory database)
```

```
to be removed from the Inventory database
...
...
DN or name of the inventoried server (as stored in the Inventory database)
to be removed from the Inventory database
```

A sample 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 "Forming the Query and Setting the Filter Conditions" on page 839.

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 scan data, 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.

Using the Inventory Removal Service for Synchronization

The Inventory Removal Service automatically removes inventoried servers from the Inventory database when the corresponding server objects are removed from eDirectory.

Sometimes it is possible that the inventoried servers in eDirectory and in the Inventory database are not synchronized because of one or both of the following reasons:

- If you kill the Inventory Service Manager, remove some server objects in the eDirectory, and restart the Inventory Service Manager.
- If you restart a previous version of the Inventory database containing servers that have already been deleted from eDirectory.

If this happened, you can use the Inventory Removal Service to remove unwanted inventoried servers from the Inventory database so the database is once again synchronized with eDirectory.

If you know the fully qualified DN names of the inventoried servers, you can specify the DN names of these servers in the inventoryremovallist.txt file.

To find the server objects that were removed from eDirectory:

1. Export the list of server objects attached to the given Inventory server using an eDirectory tool like NDSREPAIR. The eDirectory tools can be downloaded from the Cool Solutions Web site (http://www.novell.com/coolsolutions/freetools.html).
2. To export all the server objects into a .csv file, use the Data Export Wizard.

   NOTE: While exporting all the inventoried servers to a .csv files, it is necessary to select the attributes.

   The exported .csv file will contain the DNS name and the selected attributes of the inventoried servers. However, you must remove attribute values and the double-quote characters from the .csv file.
3. Compare the eDirectory exported file and the .csv file using the file comparison utility to identify the inventoried servers that do not match the .csv file.

   NOTE: The eDirectory output file and the .csv file should be in the same format for the comparison to succeed.
4. After identifying the inventoried servers that are not synchronized, place the DN names of these servers in the inventoryremovallist.txt file for the Inventory Removal Service to pick them up.

Understanding the Upgrade Service

The Upgrade service runs as a service loaded by the Service Manager.

The Upgrade service corrects the Inventory database schema and data to make it compatible with ZfS 3 SP1 and ZENworks for Desktops 4. 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	Stand alone Server
Service Manager	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Scan Collector	--	Yes	Yes	Yes	--	Yes	--	Yes	Yes
Selector	--	Yes	Yes	Yes	--	Yes	--	Yes	Yes
Storer	Yes	Yes	--	Yes	--	Yes	Yes	--	Yes
Sender	--	--	Yes	Yes	Yes	Yes	Yes	Yes	--
Receiver	Yes	Yes	--	--	Yes	Yes	Yes	Yes	--
Database	Yes	Yes	--	Yes	--	Yes	Yes	--	Yes
Inventory Removal Service	Yes	Yes	Yes	Yes	--	Yes	Yes	Yes	Yes
Upgrade Service	Yes	Yes	--	Yes	--	Yes	Yes	--	Yes

Understanding the Inventory Database

ZfS 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 28, “Understanding the ZENworks for Servers Inventory Database Schema,” on page 787

Understanding ZfS Inventory Attributes

The following table lists the Server Inventory attributes that ZENworks for Servers 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

Hardware and software enumerated values are listed separately, following the table.

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
LastLoginUser.Name	Last Login User.Name	ManageWise."User".Name	String	254	User logged in last to the Primary eDirectory tree when the inventoried server was scanned
Product.Name	Applications_Name	CIM.Product.Name	String	254	Name of the software application

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
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
WinOperating System.OSType	Windows_Name	ZENworks.WINOperatingSystem.OSType	Unsigned Small Integer (enum)		Operating system name. For example, Windows NT/ Windows 2000. See "Enumeration Values for SOFTWARE-Operating Systems-Name" on page 782.
WinOperating System.Version	Windows_Version	ZENworks.WINOperatingSystem.Version	String	254	Version of the operating system
WinOperating System.Caption	Windows_Caption	ZENworks.WINOperatingSystem.Caption	String	64	Short name of the operating system. For example, Windows NT
WinOperating System.Role	Windows_Role	ZENworks.WINOperatingSystem.Role	Integer (enum)		The role of the computer system. For example, server or workstation
WinOperating System.OtherTypeDescription	Windows_Other Description	ZENworks.WINOperatingSystem.Description	String	254	More description about the operating system
WinOperating System.InstallDate	Windows_Install Date	ZENworks.ZENOperatingSystem.InstallDate	String	25	Install date of the operating system
WinOperating System.CodePage	Windows_Code Page	ZENworks.WINOperatingSystem.CodePage	String	254	Current language code page being used
WinOperating System.TotalVisibleMemorySize	Windows_Total Memory (MB)	ZENworks.WINOperatingSystem.TotalVisibleMemorySize	Integer		Total memory as reported by the Windows operating system
WinOperating System.TotalVirtualMemorySize	Windows_Total Virtual Memory (MB)	ZENworks.WINOperatingSystem.TotalVirtualMemorySize			Total virtual 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
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 that it could be read and displayed in any an appropriate date format
InventoryScanner.InventoryServer	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

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
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
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-Mouse-Name" on page 781.
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 for Servers Inventory Database Schema" on page 787.

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.PointingType	Pointing Device_Type	CIM.PointingDevice.PointingType	Integer (enum)		The pointing device type
ZENKeyboard.Numberoffunction keys	Keyboard_Numberof 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.Video BIOSElement.Manufacturer	String	254	Manufacturer of the video BIOS driver installed on the system
VideoBIOSElement.Version	Display Driver_Version	CIM.Video BIOSElement.Version	String	254	Version of the Video BIOS driver
VideoBIOSElement.Install Date	Display Driver_Install Date	CIM.Video BIOSElement.InstallDate	String	25	Video BIOS release date
VideoBIOSElement.IsShadowed	Display Driver_Is Shadowed	CIM.Video BIOSElement.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. VideoAdpater.NumberOfColorPlanes	Unsigned Integer		Number of color planes supported by the video system
VideoAdapter.CurrentVertical Resolution	Display Adapter_ Current Vertical Resolution	ZENworks. VideoAdpater.Current Vertical Resolution	Unsigned Integer		Vertical resolution of the display

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.CurrentHorizontalResolution	Display Adapter_ Current Horizontal Resolution	ZENworks. VideoAdpater.Current Horizontal Resolution	Unsigned Integer		Horizontal resolution of the display
VideoAdapter. Description	Display Adapter_ Description	ZENworks. VideoAdpater.Description	String	254	Video adapter description
VideoAdapter.MinRefreshRate	Display Adapter_ Minimum Refresh Rate	ZENworks. VideoAdpater.MinRefresh Rate	Unsigned Integer		Minimum refresh rate of the monitor for redrawing the display, measured in Hertz
VideoAdapter.MaxRefreshRate	Display Adapter_ Maximum Refresh Rate	ZENworks. VideoAdpater.MaxRefresh Rate	Unsigned Integer		Maximum refresh rate of the monitor for redrawing the display, measured in Hertz
VideoAdapter.VideoArchitecture	Display Adapter_ Video Architecture	ZENworks. VideoAdpater.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 781.
VideoAdapter.VideoMemory Type	Display Adapter_ Video Memory Type	ZENworks. VideoAdpater.VideoMemory Type	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.
VideoAdapter.Maxmemorysupported	Display Adapter_ Maximum Memory Supported(KB)	ZENworks. VideoAdpater.MaxMemory Supported	Unsigned Integer		Maximum memory that the display adapter supports for VIDEO RAM
VideoAdapter.CurrentBitsPer Pixel	Display Adapter_ Current Bits/Pixel	ZENworks. VideoAdpater.CurrentBits PerPixel	Unsigned Integer		Number of adjacent color bits for each pixel
VideoAdapter.ChipSet	Display Adapter_ Chip Set	ZENworks. VideoAdpater.ChipSet	String	254	The chip set used in the video adapter

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.DACType	Display Adapter_ DAC Type	ZENworks. VideoAdpater.DAC Type	String	254	The digital to analog converter type used in the video adapter
VideoAdapter.ProviderName	Display Adapter_ Provider	ZENworks.VideoAdapter.Pr ovider	String	254	The manufacturer or the provider name
ZENPOTSModem.Caption	Modem_Caption	ZENworks.ZENPOTSMode m. Caption	String	64	The short name of the modem.
ZENPOTSModem.Descriptio n	Modem_Description	ZENworks.ZENPOTSMode m. Description	String	254	The complete description of the modem. For example, Standard 2400 bps modem, IBM PCMCIA HPC modem.
ZENPOTSModem.Name	Modem_Name	ZENworks.ZENPOTSMode m.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.ProviderN ame	Modem_Provider	ZENworks.ZENPOTSMode m.Provider	String	254	The manufacturer or the provider name
ZENPOTSModem.DeviceID	Modem_Device ID	ZENworks.ZENPOTSMode m.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
BIOS.PrimaryBIOS	BIOS_Primary Bios	ZENworks. BIOS.PrimaryBIOS	BIT (Used for Boolean condition s here)		True when set to 1, indicates that this BIOS is the primary BIOS. Used in systems with additional BIOS chips.

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.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.CurrentClockSpeed	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 781.
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.

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.UpgradeMethod	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 782.
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 for Servers Inventory Database Schema" on page 787.

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.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 for Servers Inventory Database Schema" on page 787.
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.Replacement Policy	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. See "Enumeration Values for HARDWARE-Memory-Cache Memory-Replacement Policy" on page 782.
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

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
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
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. For exmaple C:, A:, etc.
LocalFileSystem.FileSystem Size	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.VolumeSerial Number	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

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
LocalFileSystem.FileSystem Type	Logical Disk Drive_ File System Type	CIM.LocalFileSystem.FileS ystemType	String	254	The file system on the drive. For example, FAT, NTFS, etc.
CDROMDrive.Manufacturer	CDROM_Manufactu rer	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
CDROMDrive.Description	CDROM_ Description	ZENworks. Physical CDROM. Description	String	254	Description of the CD-ROM 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-ROM 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

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
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 Serial Port class. For more information on how they are associated, see Chapter 28, “Understanding the ZENworks for Servers Inventory Database Schema,” on page 787.
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 condition s here)		If True or 1, then it means that DMA is channel is allocated for bulk data transfer for use with devices connected to the parallel ports

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
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 28, “Understanding the ZENworks for Servers Inventory Database Schema,” on page 787.
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.Description	String	254	Description of the bus.
Bus.BusType	Bus_Bus Type	ZENworks.Bus.BusType	Integer (enum)		The bus type of the system
Bus.Name	Bus_Name	ZENworks.Bus.Name	String	254	Name of the internal system bus
Bus.DeviceID	Bus_Device ID	ZENworks.Bus.DeviceID	String	64	The unique ID for the specific bus
ZENNetworkAdapter.Name	Network Adapter_ Name	CIM.ZENworks.ZENAdapte r.Name	String	254	Network adapters installed on the system
ZENNetworkAdapter.MaxSpe ed	Network Adapter_Max_Spee d (Mbps)	CIM.ZENworks.ZENAdapte r. MaxSpeed	Unsigned Integer		Rate at which the adapter can transfer data
ZENNetworkAdapter.Perman entAddress	Network Adapter_ Permanent Address	CIM.ZENworks.ZENAdapte r. PermanentAddress	String	64	Machine address stored permanently in the adapter (MAC address)
ZENNetworkAdapter.MACAd dress	Network Adapter_ Address	CIM.ZENworks.ZENAdapte r. MACAddress	String	64	The MAC address stored in the network adapter
ZENNetworkAdapter.Provide rName	Network Adapter_ Provider	CIM.ZENworks.ZENAdapte r. Provider	String	254	The manufacturer or the provider

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
ZENNetworkAdapter.Adapter Type	Network Adapter_ Adapter Type	CIM.ZENworks.ZENAdapte r. AdapterType	String	254	Type of the adapter such as Ethernet or FDDI adapter
SoundAdapter.Description	Multimedia Card_ Description	ZENworks. SoundAdapter. Description	String	254	Description of the multimedia component for the inventoried server
SoundAdapter.Name	Multimedia Card_ Name	ZENworks. SoundAdpater. Name	String	254	Name of the sound card installed on the system
SoundAdapter.Manufacturer	Multimedia Card_ Manufacturer	ZENworks. SoundAdapter. Manufacturer	String	254	Vendor name
SoundAdapter.ProviderName	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
Battery.Chemistry	Battery_Chemistry	CIM.Battery. Chemistry	Unsigned Small Integer		Indicates battery's chemistry, such as lead acid, nickel cadmium and others. See "Enumeration Values for HARDWARE-Battery-Chemistry" on page 781.
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

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.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.TotalOutputPo wer	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.DNSName. Label	String	254	DNS name of the inventoried server
IPXProtocolEndPoint.Addres s	IPX Address_ Address	CIM.IPX Protocol Endpoint. Address	String	254	IPX address of the inventoried server
LANEndPoint. MACAddress	MAC Address_ Address	CIM.LAN Endpoint. MACAddress	String	12	MAC address of the inventoried server
MotherBoard.Version	MotherBoard_ Version	ZENworks.Motherboard.Ver sion	String	64	Motherboard version
MotherBoard.Description	MotherBoard_ Description	ZENworks.Motherboard.De scription	String	254	The description of the motherboard
MotherBoard.Manufacturer	MotherBoard_ Manufacturer	ZENworks.Motherboard.Ma nufacturer	String	254	The manufacturer of the motherboard
MotherBoard.NumberOfSlots	MotherBoard_ Number Of Slots	ZENworks.Motherboard.Nu mberofslots	Integer		The number of expansion slots on the motherboard
IRQ.Number	IRQ_IRQ Number	CIM.IRQ.IRQNumber	Unsigned Integer		The system interrupt number

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
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
DMA.DMAChannel	DMA_DMA Channel Number	CIM.DMA. DMAChannel	Unsigned Integer		The DMA channel number

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.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 or not: 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
NetWareOperatingSystem.Version	NetWare.Version	ZENworks.NetWareOperating.Version	String	254	Version of the NetWare operating system
NetWareOperatingSystem.CodePage	NetWare.Code Page	ZENworks.NetWareOperatingSystem.CodePage	String	254	The operating system language code page setting.
NetWareOperatingSystem.Caption	NetWare.Caption	ZENworks.NetWareOperatingSystem.Caption	String	64	Short name for the NetWare server. For example, NetWare
NetWareOperatingSystem.InstallDate	NetWare.Install Date	ZENworks.NetWareOperatingSystem.InstallDate	String	25	The install date of the NetWare
NetWareOperatingSystem.AccountingVersion	NetWare.Accounting Version	ZENworks.NetWareOperatingSystem.AccountingVersion	String	254	The version of the NetWare Accounting system
NetWareOperatingSystem.InternetBridgeSupport	NetWare.Internet Bridge Support	ZENworks.NetWareOperatingSystem.MaxNumberOfConnections	String	254	Maximum number of connections allowed
NetWareOperatingSystem.MaxNumberOfConnections	NetWare.Maximum Number Of Connections	ZENworks.NetWareOperatingSystem.MaxNumberOfConnections	Integer		Maximum number of connections allowed
NetWareOperatingSystem.MaxNumberOfVolumes	NetWare.Maximum Number Of Volumes	ZENworks.NetWareOperatingSystem.MaxNumberOfVolumes	Integer		Maximum number of volumes allowed

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the .CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
NetWareOperatingSystem.PeakConnectionsUsed	NetWare.Peak Connections Used	ZENworks.NetWareOperatingSystem.PeakConnectionsUsed	Integer		Maximum number of connections used
NetWareOperatingSystem.PrintServerVersion	NetWare.Print Server Version	ZENworks.NetWareOperatingSystem.PrintServerVersion	String	254	The print server version if the server is used as a print server
NetWareOperatingSystem.QueuingVersion	NetWare.Queuing Version	NetWareOperatingSystem.QueuingVersion	String	254	The NetWare Print server’s print queue version if this server is used as a print server
NetWareOperatingSystem.RevisionLevel	NetWare.Revision Level	ZENworks.NetWareOperatingSystem.RevisionLevel	String	254	The revision level of NetWare
NetWareOperatingSystem.SecurityRestrictionLevel	NetWare.Security Restriction Level	ZENworks.NetWareOperatingSystem.SecurityRestrictionLevel	String	254	The security restriction level
NetWareOperatingSystem.SFTLevel	NetWare.SFT Level	ZENworks.NetWareOperatingSystem.SFTLevel	String	254	The SFT level
NetWareOperatingSystem.TTSLevel	NetWare.TTS Level	ZENworks.NetWareOperatingSystem.TTSLevel	String	254	The TTS level
NetWareOperatingSystem.VAPVersion	NetWare.VAP Version	ZENworks.NetWareOperatingSystem.VAPVersion	String	254	The VAP version
NetWareOperatingSystem.VirtualConsoleVersion	NetWare.Virtual Console Version	ZENworks.NetWareOperatingSystem.VirtualConsoleVersion	String	254	The virtual console version
NetWareOperatingSystem.InternalNetworkNumber	NetWare.Internal Network Number	ZENworks.NetWareOperatingSystem.InternalNetworkNumber	String	254	The internal network number reported by Netware
NetWareOperatingSystem.TotalVisibleMemorySize	NetWare.Total Memory(MB)	ZENworks.NetWareOperatingSystem.TotalVisibleMemorySize	String	254	The total primary memory as reported by NetWare
NetWareOperatingSystem.TotalVirtualMemorySize	NetWare.Total Virtual Memory(MB)	ZENworks.NetWareOperatingSystem.TotalVirtualMemorySize	String	254	The total virtual memory size used by NetWare

Enumeration Values for HARDWARE-Display Adapter.Video Architecture

The enumeration values are:

1 = "Other"	6 = "SVGA"	11 = "XGA"
2 = "Unknown"	7 = "MDA"	12 = "Linear Frame Buffer"
3 = "CGA"	8 = "HGC"	160 = "PC-98"
4 = "EGA"	9 = "MCGA"	
5 = "VGA"	10 = "8514A"	

Enumeration Values for HARDWARE-Display Adapter.Video Memory Type

The enumeration values are:

1 = "Other"	6 = "WRAM"	11 = "3DRAM"
2 = "Unknown"	7 = "EDO RAM"	12 = "SDRAM"
3 = "VRAM"	8 = "Burst Synchronous DRAM"	13 = "SGRAM"
4 = "DRAM"	9 = "Pipelined Burst SRAM"	
5 = "SRAM"	10 = "CDRAM"	

Enumeration Values for HARDWARE-Mouse-Name

The enumeration values are:

1 = "Other"	4 = "Track Ball"	7 = "Touch Pad"
2 = "Unknown"	5 = "Track Point"	8 = "Touch Screen"
3 = "Mouse"	6 = "Glide Point"	9 = "Mouse - Optical Sensor"

Enumeration Values for HARDWARE-Battery-Chemistry

The enumeration values are:

1 = "Other"	5 = "Nickel Metal Hydride"
2 = "Unknown"	6 = "Lithium-ion"
3 = "Lead Acid"	7 = "Zinc air"
4 = "Nickel Cadmium"	8 = "Lithium Polymer"

Enumeration Values for HARDWARE-Processor-Processor Family

The enumeration values are:

1 = "Other"	15 = "Celeron(TM)"	130 = "Itanium(TM) Processor"
2 = "Unknown"	16 = "Pentium(R) II Xeon(TM)"	176 = "Pentium(R) III Xeon(TM)"
11 = "Pentium(R) Brand"	17 = "Pentium(R) III"	177= "Pentium(R) III Processor with Intel(R) SpeedStep(TM) Technology"

12 = "Pentium(R) Pro"	24 = "AMD Duron(TM) Processor Family"	178 = "Pentium(R) 4 Processor"
13 = "Pentium(R) II"	29 = "AMD Athlon(TM) Processor Family"	
14 = "Pentium(R) Processor with MMX(TM) Technology"	30 = "AMD29000 Family"	

Enumeration Values for HARDWARE-Processor-Upgrade Method

The enumeration values are:

1= "Other"	5 = "Replacement/Piggy Back"	9 = "Slot 2"
2 = "Unknown"	6 = "None"	10 = "370 Pin Socket"
3 = "Daughter Board"	7 = "LIF Socket"	11 = "Slot A"
4 = "ZIF Socket"	8 = "Slot 1"	12 = "Slot M"

Enumeration Values for HARDWARE-Memory-Cache Memory-Replacement Policy

The enumeration values are:

1 = "Other"	5 = "Last In First Out (LIFO)"
2 = "Unknown"	6 = "Least Frequently Used (LFU)"
3 = "Least Recently Used (LRU)"	7 = "Most Frequently Used (MFU)"
4 = "First In First Out (FIFO)"	8 = "Data Dependent Multiple Algorithm"

Enumeration Values for SOFTWARE-Operating Systems-Name

The enumeration values are:

0 = "Unknown"	17 = "WIN98"	58 = "Windows 2000"
1 = "Other"	18 = "WINNT"	59 = "Dedicatedo"
16 = "WIN95"	21 = "NetWare"	63 = "Windows (R) Me"

Enumeration Values for HARDWARE-Bus-Protocol Supported

The enumeration values are:

0 = "Internal"	6 = "VME Bus"	12 = "Internal Processor"
1 = "ISA"	7 = "NuBus"	13 = "Internal Power Bus"
2 = "EISA"	8 = "PCMCIA Bus"	14 = "PNP ISA Bus"
3 = "MicroChannel"	9 = "C Bus"	15 = "PNP Bus"
4= "TurboChannel""	10 = "MPI Bus"	16= "Maximum Interface Type"
5 = "PCI Bus"	11 = "MPSA Bus"	

Enumeration Values for GENERAL-Asset-Management Technology

The enumeration values are:

1 = "Unknown"	3= "DMI Enabled"	5= "SNMP Enabled"
2 = "Other"	4 = "WMI Enabled"	6 = "DMI and WMI Enabled"

Enumeration Values for SOFTWARE-Operating Systems-Windows-Role

The enumeration values are:

0 = "Unknown"	2 = "Managed Servers"
1 = "Other"	3 = "Managed Workstation"

Enumeration Values for SOFTWARE-Scanner Information-Scan Mode

The enumeration values are:

1 = "Unknown"	3= "DMI "	5= "SNMP"
2 = "Other"	4 = "WMI "	6 = "DMI & WMI "

Enumeration Values for HARDWARE-Processor-Role

The enumeration values are:

1 = "Other"	3= "Central Processor "	5= "DSP Processor"
2 = "Unknown"	4 = "Math Processor "	6 = "Video Processor "

Enumeration Values for HARDWARE-Processor-Upgrade Method

The enumeration values are:

1 = "Other"	5 = "Replacement/Piggy Back "	9 = "Slot 2"
2 = "Unknown"	6 = "None "	10 = "370 Pin Socket"
3 = "Daughter Board"	7= "LIF Socket"	11 = "Slot A"
4= "ZIF Socket "	8 = "Slot 1"	12 = "Slot M"

Enumeration Values for SYSTEM-Cache Memory-Level

The enumeration values are:

1 = "Other"	3 = "Primary "	5 = "Tertiary"
2 = "Unknown"	4 = "Secondary "	6 = "Not Applicable"

Enumeration Values for SYSTEM-Cache Memory-Level

The enumeration values are:

1 = "Other"	3 = "Write Back "	5 = "Varies with Address"
2 = "Unknown"	4 = "Write Through "	6 = "Determination Per I/O"

Enumeration Values for SYSTEM-Cache Memory-Cache Type

The enumeration values are:

1 = "Other"	3 = "Instruction "	5 = "Unified"
2 = "Unknown"	4 = "Data "	

Enumeration Values for SYSTEM-Cache Memory-Replacement Policy

The enumeration values are:

1 = "Other"	4 = "First In First Out (FIFO) "	7 = "Most Frequently Used (MFU)"
2 = "Unknown"	5 = "Last In First Out (LIFO) "	8="Data Dependent Multiple Algorithms"
3 = "Least Recently Used (LRU)"	6 = "Least Frequently Used (LFU) "	

Enumeration Values for SYSTEM-Cache Memory-Read Policy

The enumeration values are:

1 = "Other"	3 = "Read "	5 = "Read and Read-ahead"
2 = "Unknown"	4 = "Read-ahead "	6="Determination Per I/O"

Enumeration Values for SYSTEM-Cache Memory-Associativity

The enumeration values are:

1 = "Other"	4 = "2-way Set-Associative "	7 = "8-way Set-Associative"
2 = "Unknown"	5 = "4-way Set-Associative"	8="16-way Set-Associative"
3 = "Direct Mapped"	6 = "Fully Associative"	

Enumeration Values for SYSTEM-IRQ-Availability

The enumeration values are:

1 = "Other"	3 = "Available "	5 = "In Use and Available/ Shareable"
2 = "Unknown"	4 = "In Use/Not Available "	

Enumeration Values for SYSTEM-IRQ-IRQ Trigger Type

The enumeration values are:

1 = "Other"	3 = "Level "
2 = "Unknown"	4 = "Edge "

Enumeration Values for SYSTEM-DMA-Availability

The enumeration values are:

1 = "Other"	3 = "Available "	5 = "In Use and Available/ Shareable"
2 = "Unknown"	4 = "In Use/Not Available "	

28 Understanding the ZENworks for Servers Inventory Database Schema

This section describes the design of the Novell® ZENworks® for Servers (ZfS) 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" on page 787
- "CIM Schema" on page 788
- "Inventory Database Schema in ZfS" on page 798

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®, UNIX*, 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

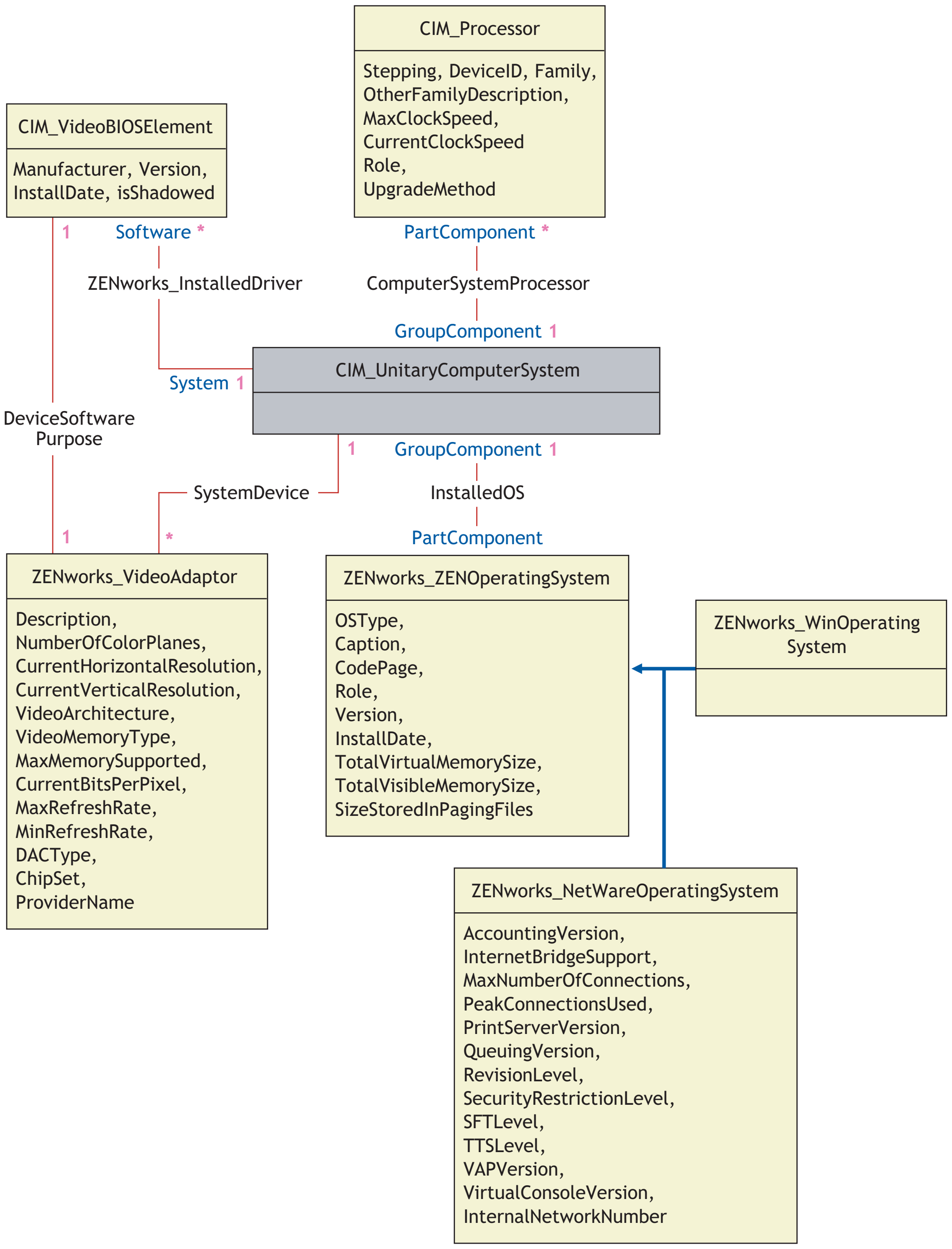
The elements of the meta schema are classes, properties, and methods. The meta schema also supports indications and associations as types of classes and references as types of properties.

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.

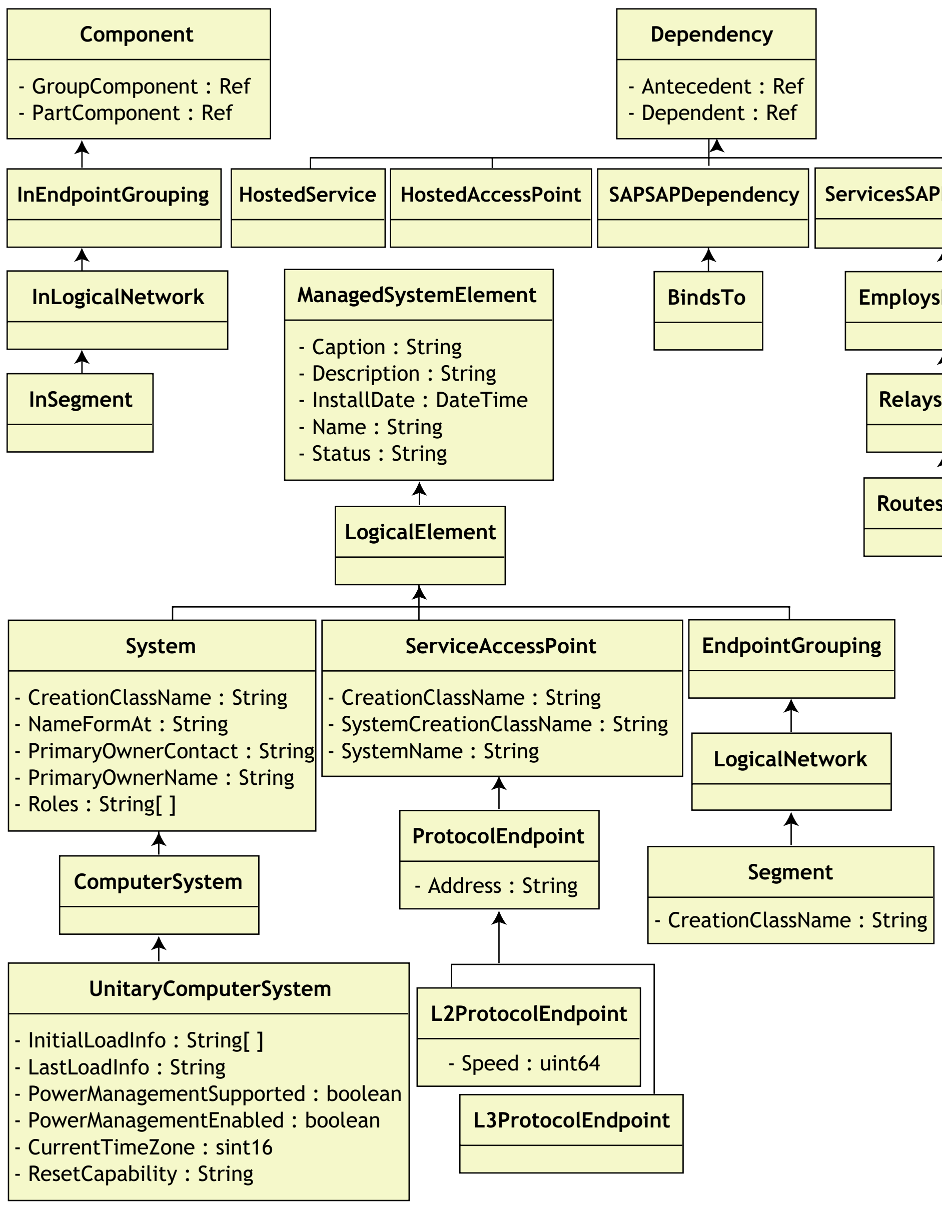
The schema fragment in the following illustration shows the relationships between some CIM objects that ZfS uses.



CIM_Processor
Stepping, DeviceID, Family, OtherFamilyDescription, MaxClockSpeed, CurrentClockSpeed Role, UpgradeMethod
CIM_VideoBIOSElement
Manufacturer, Version, InstallDate, isShadowed
1
Software *
ZENworks_InstalledDriver
PartComponent *
ComputerSystemProcessor
GroupComponent 1
System 1
CIM_UnitaryComputerSystem
DeviceSoftware Purpose
1
GroupComponent 1
SystemDevice
InstalledOS
PartComponent
1
*
ZENworks_VideoAdaptor
Description, NumberOfColorPlanes, CurrentHorizontalResolution, CurrentVerticalResolution, VideoArchitecture, VideoMemoryType, MaxMemorySupported, CurrentBitsPerPixel, MaxRefreshRate, MinRefreshRate, DACType, ChipSet, ProviderName
ZENworks_ZENOperatingSystem
OSType, Caption, CodePage, Role, Version, InstallDate, TotalVirtualMemorySize, TotalVisibleMemorySize, SizeStoredInPagingFiles
ZENworks_WinOperating System
ZENworks_NetWareOperatingSystem
AccountingVersion, InternetBridgeSupport, MaxNumberOfConnections, PeakConnectionsUsed, PrintServerVersion, QueuingVersion, RevisionLevel, SecurityRestrictionLevel, SFTLevel, TTSLevel, VAPVersion, VirtualConsoleVersion, InternalNetworkNumber

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:

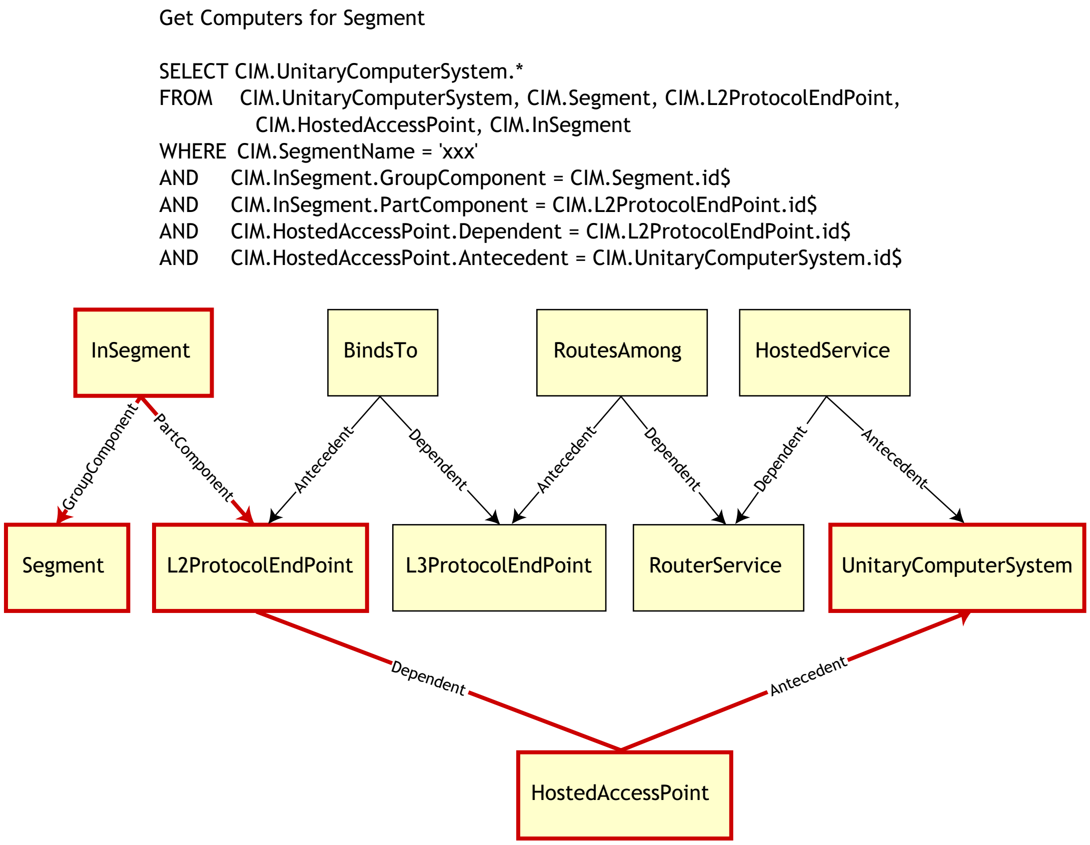
- **Logical Schema:** The logical schema defines how the data appears to applications, similar to an API. The goal is that the logical schema remains the same irrespective of the underlying database so that application software can run unchanged on any supported databases. Though SQL (pronounced as sequel) 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 data 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 792
- "Physical Schema" on page 798

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.

Typically, each CIM class has the following:

- A constructor procedure to generate an instance of the class. For more information, see "Constructor" on page 796.
- A destructor procedure to destroy an instance of the class. For more information, see "Destructor" on page 798.
- A view to access and update the values of properties of the class.

ZfS 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 792
- "Users and Roles" on page 793
- "Data Types" on page 793
- "Views" on page 794
- "Object Identifier Id$" on page 795
- "Constructor" on page 796
- "Destructor" on page 798

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 ZfS 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 and MWO_Reader or MWO_Updater for an Oracle 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 ZfS, 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.

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. For example, the CIM class CIM_System represents a SQL view named CIM.System, as shown in the following illustration.

The CIM.System view is created with attributes that are selected from multiple tables. These attributes include: id$ selected from cim.t$ManagedSystemElement,class$ is filled up automatically using the function mw_dba.extractClass, Caption selected from cim.t$ManagedSystemElement, Description selected from cim.t$ManagedSystemElement, InstallDate selected from cim.t$ManagedSystemElement, Status selected from cim.t$ManagedSystemElement, CreationClassName selected from cim.t$System, Name selected from cim.t$ManagedSystemElement. NameFormat selected from cim.t$System.NameFormat, PrimaryOwnerContact selected from cim.t$System, and PrimaryOwnerName selected from cim.t$System. The view is created by joining the tables CIM.t$ManagedSystemElement and CIM.t$System where the id$ of both the tables are same.

The CIM.SYSTEM view is as follows:

```
CREATE VIEW CIM.System

{

  id$,

  class$,

  Caption,

  Description,

  InstallDate,

  Status,

  CreationClassName,

  Name,

  NameFormat,

  PrimaryOwnwerContact,

  PrimaryOwnerName

}

AS SELECT

  CIM.t$ManagedSystemElement.id$

  MW_DBA.extractClass(CIM.t$ManagedSystemElement.id$),

  CIM.t$ManagedSystemElement.Caption,

  CIM.t$ManagedSystemElement.Description,

  CIM.t$ManagedSystemElement.InstallDate,

  CIM.t$ManagedSystemElement.Status,

  CIM.t$System.CreationClassName,

  CIM.t$ManagedSystemElement.Name,
  CIM.t$System.NameFormat,
  CIM.t$System.PrimaryOwnerContact,
  CIM.t$System.PrimaryOwnerName
FROM
  CIM.t$ManagedSystemElement,
  CIM.t$System
WHERE
  CIM.t$ManagedSystemElement.id$ = CIM.t$System.id$
```

In addition to the properties of the class, the view has the following two additional fields:

- **Id$:** An object identifier that uniquely identifies the particular instance of the class. See "Object Identifier Id$" on page 795.
- **Class$:** An integer field that identifies the actual type of the class. For example, the actual type of a CIM_System can be any of the concrete subclasses of CIM_System.

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.

Object Identifier Id$

Id$ is a 64-bit object identifier that uniquely identifies a particular instance of a class. This object identifier is usually used as an opaque handle to a particular instance. Id$ is modeled as a signed number for ease of manipulation in Java as a long data type.

Id$ contains the following three parts of information, which can each be extracted by invoking the appropriate stored procedure.

- The most significant 16 bits of id$ encode the actual class of the object.

  This field can be extracted using the MW_DBA.extractClass() function. This field is used for type decisions or to access additional information about the class from the MW_DBA.Class table.
- The next 8 bits of id$ encode the site ID.

  The site ID uniquely identifies the database on a particular site. This field makes the object identifier unique across as many as 256 sites so that inventory data from multiple sites can be rolled up into a single database (Root Server with Database) for querying and reporting without causing key conflicts. The site ID can be extracted using the MW_DBA.extractSite() function.
- The least significant 40 bits uniquely identify the particular instance of that class.

  This part can be extracted using the MW_DBA.extractId() function. This is not useful from an end-user's perspective.

The id$ field is used in its entirety as an opaque handle to an instance of a class. When an association class represents a relationship between instances of two classes, the reference fields of the association hold the id$ of the referenced instances (like the pointers). Therefore, id$ and these reference fields are frequently used in Join conditions when constructing the database queries that reference more than one view.

Constructor

Each concrete (non-abstract) CIM class has a constructor stored procedure that must be called to create an instance of the class. This stored procedure has input parameters that allow the user to specify a value for each property in the class, and a single output parameter that returns the id$ allocated to the created instance. The application uses this returned id$ value to construct association classes that reference that particular instance.

The constructor is named by prefixing the root name with c$, and each parameter is named by prefixing the root property name with p$. For example, the constructor for CIM_UnitaryComputerSystem, a subclass of CIM_System, is named CIM.c$UnitaryComputerSystem and is constructed for Oracle as shown in the following example:

```
CREATE PROCEDURE CIM.c$UnitaryComputerSystem

(

p$id$  OUT NUMBER,

p$Caption IN CIM.t$ManagedSystemElement.Caption%TYPE DEFAULT  NULL,

p$Description IN CIM.t$ManagedSystemDescription%TYPE DEFAULT NULL,

p$InstallDate IN CIM.t$ManagedSystemElement.InstallDate%TYPE DEFAULT NULL,

p$Status IN CIM.t$ManagedSystemElement.Status%TYPE DEFAULT NULL,

p$CreationClassName IN CIM.t$System.CreationClassName%TYPE DEFAULT NULL,

p$Name IN CIM.t$ManagedSystemElement.Name%TYPE DEFAULT NULL,

p$PrimaryOwnerContact IN CIM.t$System.PrimaryOwnerContact%TYPE DEFAULT NULL,

p$PrimaryOwnerName IN CIM.t$System.PrimaryOwnerName%TYPE DEFAULT NULL,

p$NameFormat IN CIM.t$System.NameFormat%TYPE DEFAULT NULL,

p$LastLoadInfo IN CIM.t$UnitaryComputerSystem.LastLoadInfo%TYPE DEFAULT
NULL,

p$ResetCapability IN CIM.t$UnitaryComputerSystem.ResetCapability%TYPE
DEFAULT NULL,

p$PowerManagementSupported IN
CIM.t$UnitaryComputerSystem.PowerManagementSupported%TYPE DEFAULT NULL,

p$PowerState IN CIM.t$UnitaryComputerSystem.PowerState%TYPE DEFAULT NULL

)IS

  temp NUMBER;

BEGIN

  LOOP

  SELECT CIM.s$UnitaryComputerSystem.NEXTVAL INTO temp FROM DUAL;

  SELECT MW_DBA.makeId(240, temp) INTO temp FROM DUAL;

  EXIT WHEN MOD(temp,100) != 0;

  END LOOP;

  p$id$ := temp;
```

```
INSERT INTO CIM.t$ManagedSystemElement (id$, classOid$,
Caption, Description, InstallDate, Status, Name)VALUES(p$id$,
HEXTORAW('0302100203'), p$Caption, p$Description,
p$InstallDate, p$Status, p$Name);

INSERT INTO CIM.t$System (id$, CreationClassName,
PrimaryOwnerContact, PrimaryOwnerName,
NameFormat)VALUES(p$id$, p$CreationClassName,
p$PrimaryOwnerContact, p$PrimaryOwnerName, p$NameFormat);

INSERT INTO CIM.t$UnitaryComputerSystem (id$, LastLoadInfo,
ResetCapability, PowerManagementSupported, PowerState)
VALUES (p$id$, p$LastLoadInfo,
p$ResetCapability,p$PowerManagementSupported, p$PowerState);

END;
```

Stored procedures can be called with either positional arguments or keyword arguments, or with a combination of the two. If any positional arguments are supplied, they must precede any keyword arguments. Always use keyword arguments when calling constructor stored procedures. This provides better insulation from CIM schema changes that cause either the insertion of extra parameters or the recording of existing parameters, either of which can break a positional call in a possible undetectable way. The procedures are generated such that any omitted parameters will default to NULL.

It is permissible to use the positional notation for the first parameter p$id$, which is the output parameter that returns the object identifier of the newly created instance.

The following code sample shows how to call a stored procedure using positional notation for the first argument and keyword notation for all subsequent arguments on Sybase.

```
CallableStatement CS =

conn.prepareCall( "{call CIM.c$UnitaryComputerSystem( ?,  p$Name=?,
p$Description=?)}" )

cs.registerOutParameter ( 1, java.sql.Types.BIGINT ); //id$

cs.setString( 2, "Bogus_UCS_1") ; //Name

cs.setString( 3, "Created with mixture of positional & keyword args" ); //
Description

cs.executeUpdate();

long id = cs.getLong ( 1 );

SQLWarning w = cs.getWarnings();

if( w != null )

  printWarnings( w );

else

  System.out.println("Created UCS id$ = " + id );
```

The syntax for keyword notation differs in Sybase ASA and Oracle. In Sybase ASA, the syntax is KEYWORD=*value*. In Oracle, the syntax is KEYWORD => *value*. Properly written code will dynamically construct the call string using syntax appropriate for the database in use.

Destructor

Each non-abstract CIM class has a destructor stored procedure that is called to destroy an instance of the class. This stored procedure has only one input parameter that specifies the object identifier (id$) of the instance to be destroyed and returns no value.

The destructor deletes the appropriate rows in all relevant tables, including the rows in the inheritance chain and any associations that reference the instance being destroyed. Only the association is destroyed; the associated objects associated are not destroyed. If there is need to destroy the association, the programmers must ensure that they are not destroyed.The destructor is named by prefixing the root name with d$ and the single object identifier parameter is named p$id$. This procedure is called using positional notation. For example, the destructor for CIM_UnitaryComputerSystem, a concrete subclass of CIM_System, is named as CIM.d$UnitaryComputerSystem.

Physical Schema

The physical schema comprises elements necessary to implement the database. The physical schema differs for each database. A typical physical schema consists of:

- 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 ZfS

The following section describes the database schema classes and the extensions and associations made to the CIM schema for use in ZfS. 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 ZfS 3 database schema:

- "Case Study of CIM Schema Implementation in ZfS" on page 799
- "Legends for Schema Diagrams" on page 801
- "CIM Classes and Extension Classes in ZfS" on page 801
- "Schema Diagrams of CIM and the Extension Schema in ZfS" on page 803
- "Sample Inventory Database Queries" on page 808

Case Study of CIM Schema Implementation in ZfS

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.

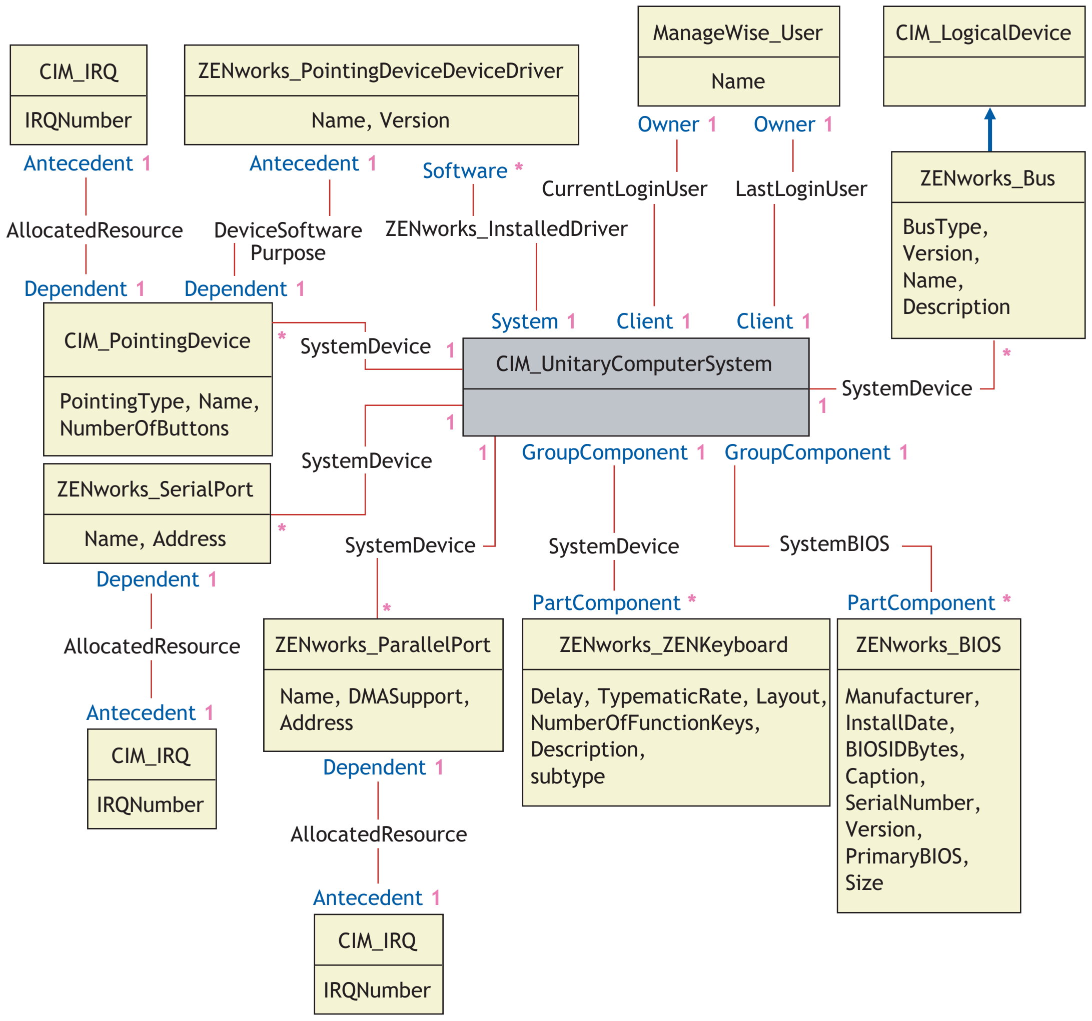
Class ZENworks.BIOS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBIOS.PartComponent pointing to ZENworks.BIOS. The relationship between the two classes is one to one. This means a computer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

Class ManageWise.Usera 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, 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 an explanation of the CIM and extended classes, see "CIM Classes and Extension Classes in ZfS" on page 801. For schema diagrams of other classes, see "Schema Diagrams of CIM and the Extension Schema in ZfS" on page 803.

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).

CIM Classes and Extension Classes in ZfS

The following table describes the CIM and extension classes that ZfS uses:

CIM and Extension Class in ZfS	Description of the details that the Class Models
CIM.PointingDevice	Any pointing device available on the managed system. Mostly used to model the mouse.
ZENworks.SystemInfo	Identification details about the system such as serial number and asset tag.
ZENworks.PointingDeviceDeviceDriver	Device driver that is installed with the pointing device.
ZENworks.SerialPort	Serial ports on the managed system.
ZENworks.ParallelPort	Parallel ports on the managed system.
ZENworks.ZENKeyboard	Attributes modeling the properties of the system keyboard.
ZENworks.BIOS	BIOS software on the system.
ZENworks.Bus	System bus in the system.
ManageWise.User	Details of the user who was logged in to the inventoried server.
ManageWise.MSDomainName	Name of the domain to which the Windows NT inventoried server is attached.
ManageWise.NDSName	DN name and tree under which the managed inventoried server is registered in Novell eDirectory™.
CIM.VideoBIOSElement:	Video driver.
CIM.Processor	Processor of the inventoried server.
ZENworks.Videoadapter	Properties of the monitor and the adapter connecting it.
ZENworks.ZENOperatingSystem	Details of the operating system.

CIM and Extension Class in ZfS	Description of the details that the Class Models
ZENworks.InventoryScanner	Details of the inventory scanner that has scanned for hardware and software details of the managed inventoried server.
ZENworks.NetwareClient	NetWare client version of the inventoried server.
CIM.Product	Software installed on the managed system. Key attributes are the names of the product, vendor, and version.
ZENworks.ZENNetworkAdapter	Information on the properties of the network adapter.
ZENworks.NetworkAdapterDriver	Network card adapter driver information.
CIM.IPProtocolEndpoint	IP address of the inventoried server.
CIM.IPXProtocolEndpoint	IPX address of the inventoried server.
CIM.LANEndpoint	Active MAC address.
ManageWise.DNSName	DNS name of the inventoried server.
ZENworks.SoundAdapter	Description of the multimedia adapter on the inventoried server.
ZENworks.ZENPOTSModem	Physical configuration of the modem device.
CIM.DMA	Information about the system DMA channels.
CIM.CacheMemory	Information about the configured system cache.
CIM.IRQ	List of Interrupt channels and their status on the system. They are also associated to devices that use the specified interrupt number.
ZENworks.MotherBoard	Information about the motherboard on the inventoried server.
CIM.PowerSupply	Information about the power supply unit of the inventoried server.
CIM.Battery	Physical details of the system battery.
CIM.Card	Details of adapter cards mounted on the system board.
CIM.Slot	Expansion slots available on the system board.
ZENworks.StoragePhysicalMedia	Physical information about the storage devices on the inventoried server, such as hard disk, floppy drives, CD drives, and others.
ZENworks.LogicalDiskette	Drive mapped to the floppy drive.
ZENworks.PhysicalDiskette	Derived from ZENworks.StoragePhysicalMedia to model the floppy disk drive.
ZENworks.PhysicalDiskDrive	Derived from ZENworks.StoragePhysicalMedia to model the hard disk.
ZENworks.LogicalDiskDrive	Information about the local drives on the hard disk.

CIM and Extension Class in ZfS	Description of the details that the Class Models
CIM.LocalFileSystem	Information about the local file system installed on the Windows servers.
ZENworks.PhysicalCDROM	Derived from ZENworks.StoragePhysicalMedia to model the CD drive.
ZENworks.WinOperatingSystem	Details of the Windows operating system.
ZENworks.NetWareOperatingSystem	Details of the NetWare operating system.
ZENworks.ZENDiskDrive	Details of fixed or removable disk drives.
ZENworks.LogicalCDROM	Drive mapped to the CD drive.

Schema Diagrams of CIM and the Extension Schema in ZfS

The following schema diagrams of the CIM and extension schema model the Inventory database in ZfS.

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.

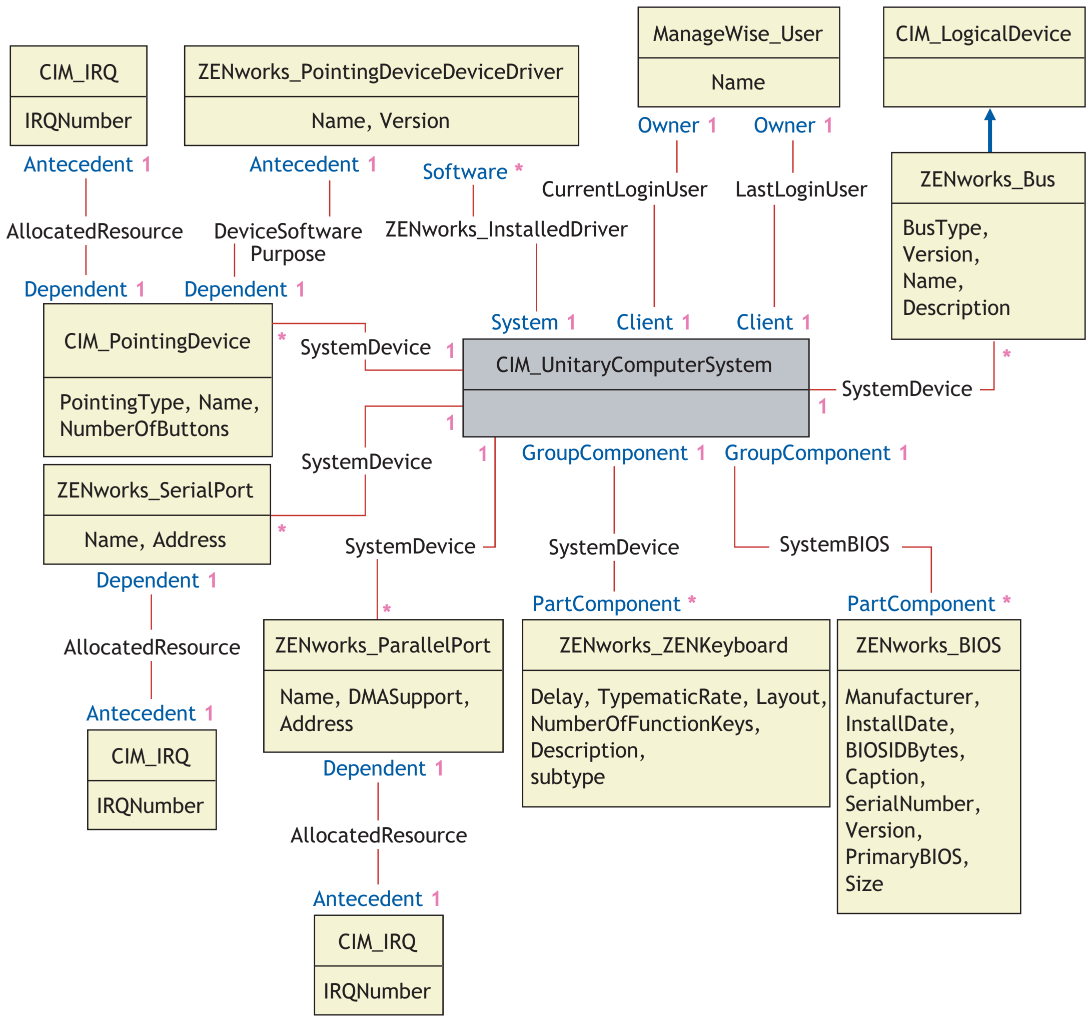
Class ZENworks.BIOS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBIOS.PartComponent pointing to ZENworks.BIOS. The relationship between the two classes is one to one. This means a computer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

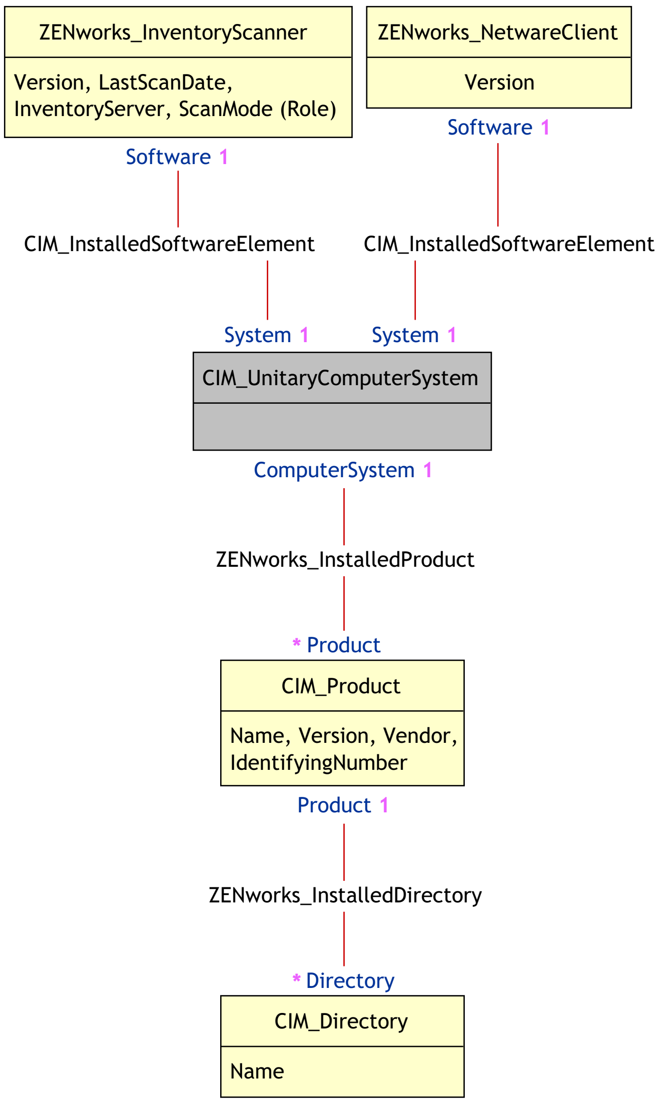
Class CIM.ZENworks.ParallelPort associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemDevice.PartComponent pointing to CIM.ZENworks.ParallelPort. The relationship between the two classes is one to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

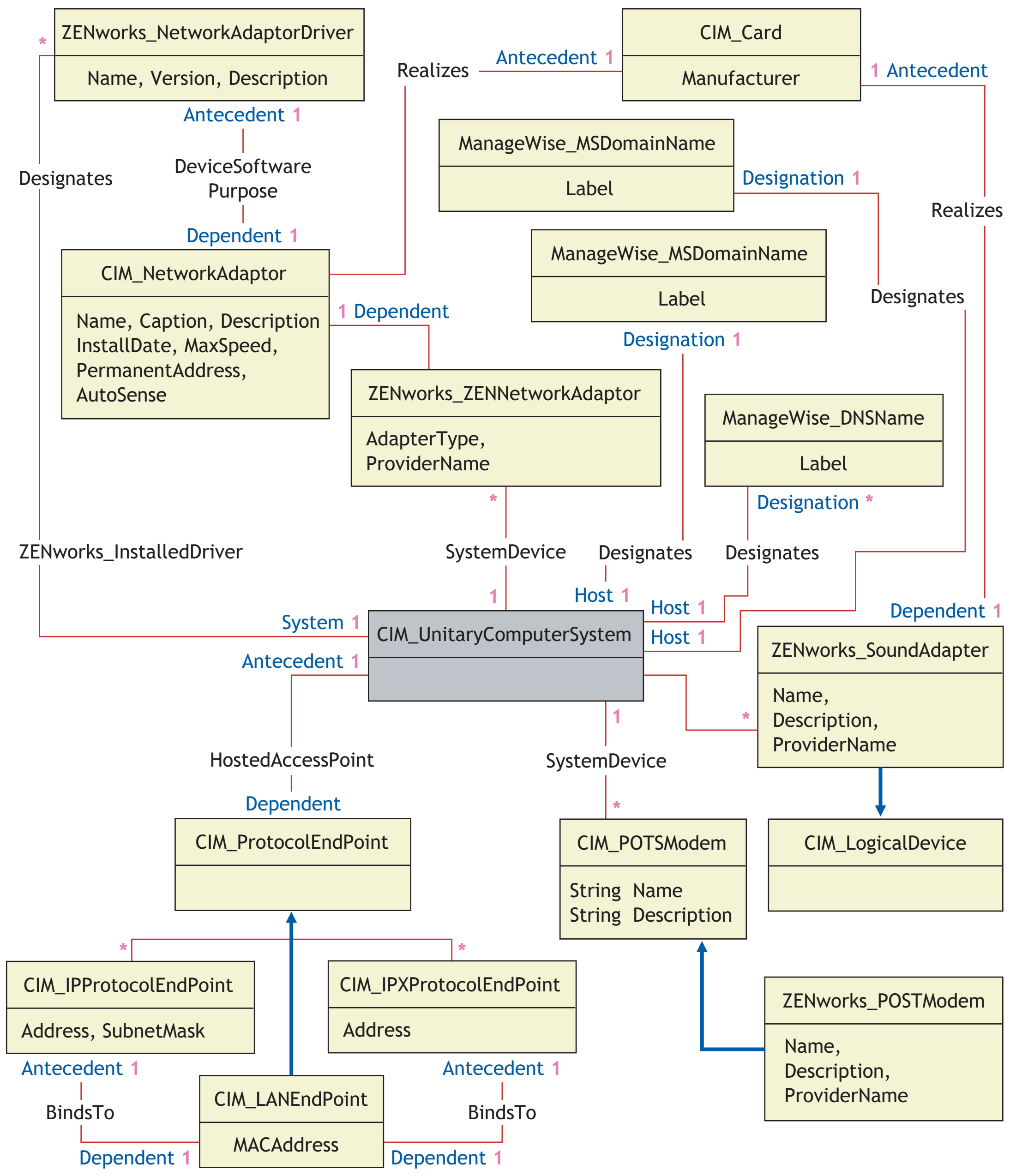
Class ZENworks.BUS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBUS.PartComponent pointing to ZENworks.BUS. The relationship between the two classes is one to one. This means a computer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

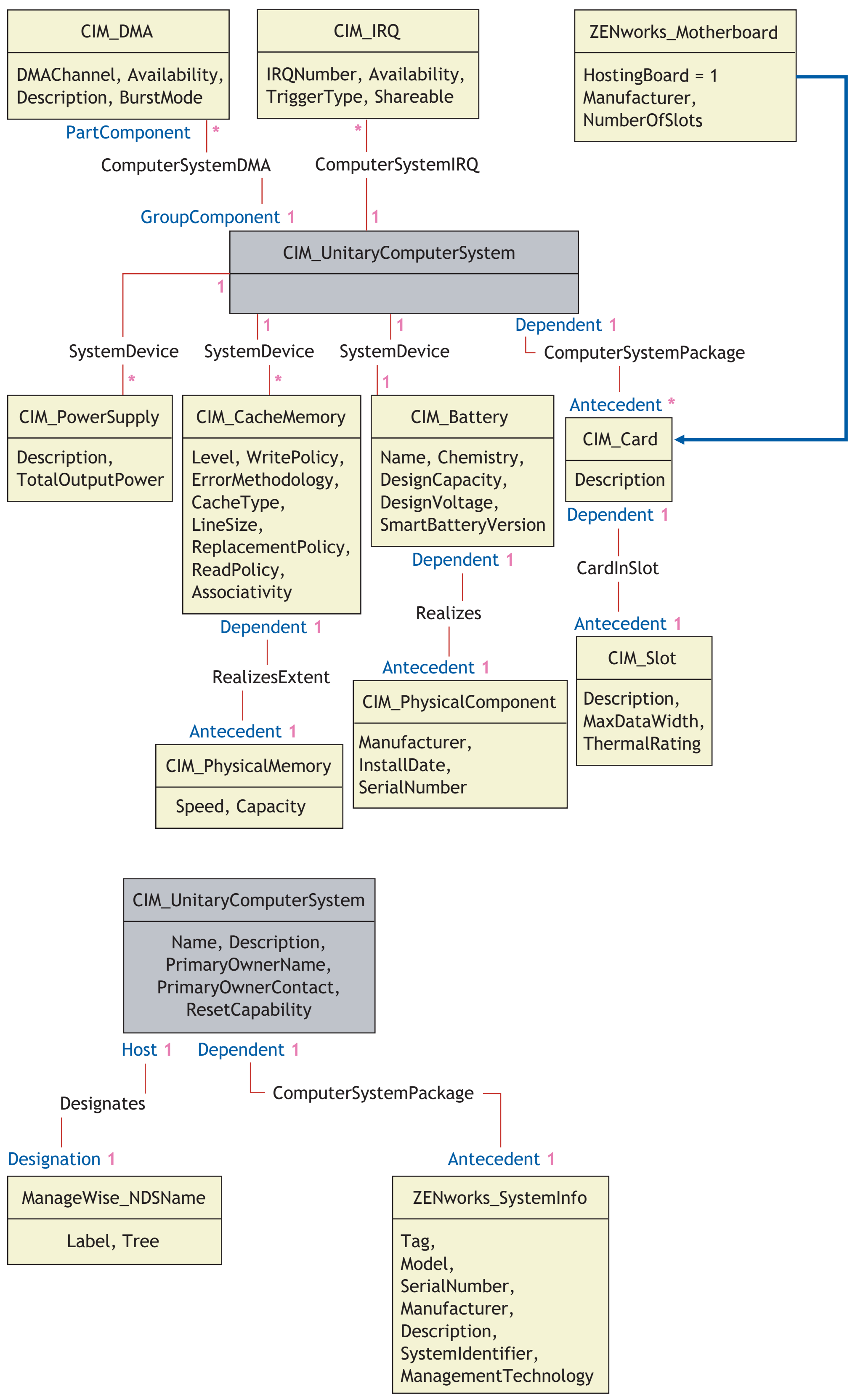
Class ManageWise.User has two associations with CIM.UnitaryComputerSystem; CurrentLoginUser and LastLoginUser. In the CurrentLoginUser association, the specific instance of User is the one who is currently logged into the inventoried server. In the LastLoginUser association, the specific instance of User is the one who logged last into the inventoried server.

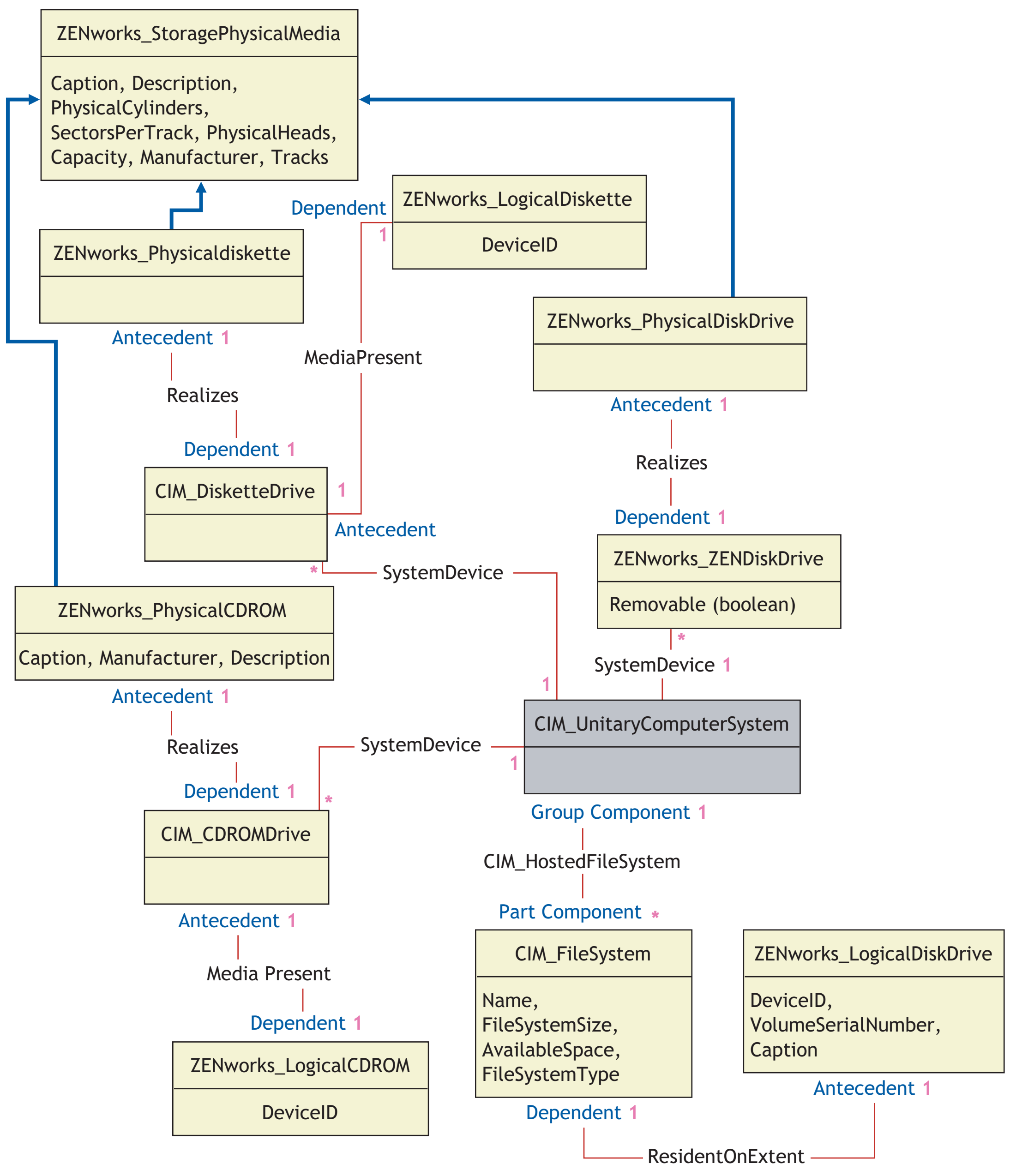
Class CIM.IRQ associates to CIM.ParallelPort using the association CIM.AllocatedResource. Dependent pointing to CIM.ParallelPort and Antecedent pointing to CIM.IRQ.




ZENworks_InventoryScanner
Version, LastScanDate, InventoryServer, ScanMode (Role)
ZENworks_NetwareClient
Version
Software 1
Software 1
CIM_InstalledSoftwareElement
CIM_InstalledSoftwareElement
System 1
System 1
CIM_UnitaryComputerSystem
ComputerSystem 1
ZENworks_InstalledProduct
* Product
CIM_Product
Name, Version, Vendor, IdentifyingNumber
Product 1
ZENworks_InstalledDirectory
* Directory
CIM_Directory
Name


ZENworks_NetworkAdaptorDriver
Name, Version, Description
CIM_Card
Manufacturer
Realizes
Antecedent 1
1 Antecedent
Antecedent 1
DeviceSoftware Purpose
Dependent 1
Designates
ManageWise_MSDomainName
Label
Designation 1
Realizes
CIM_NetworkAdaptor
Name, Caption, Description InstallDate, MaxSpeed, PermanentAddress, AutoSense
1 Dependent
ManageWise_MSDomainName
Label
Designation 1
Designates
ZENworks_ZENNetworkAdaptor
AdapterType, ProviderName
ManageWise_DNSName
Label
Designation *
ZENworks_InstalledDriver
SystemDevice
Designates
Designates
Host 1
Host 1
Host 1
System 1
CIM_UnitaryComputerSystem
Antecedent 1
Dependent 1
ZENworks_SoundAdapter
Name, Description, ProviderName
HostedAccessPoint
SystemDevice
Dependent
CIM_ProtocolEndPoint
CIM_POTSModem
String Name
String Description
CIM_LogicalDevice
CIM_IPProtocolEndPoint
Address, SubnetMask
CIM_IPXProtocolEndPoint
Address
ZENworks_POSTModem
Name, Description, ProviderName
Antecedent 1
Antecedent 1
BindsTo
CIM_LANEndPoint
MACAddress
BindsTo
Dependent 1
Dependent 1


CIM_DMA
DMAChannel, Availability, Description, BurstMode
CIM_IRQ
IRQNumber, Availability, TriggerType, Shareable
ZENworks_Motherboard
HostingBoard = 1 Manufacturer, NumberOfSlots
PartComponent *
*
ComputerSystemDMA
ComputerSystemIRQ
GroupComponent 1
1
CIM_UnitaryComputerSystem
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
Dependent 1
Dependent 1
CardInSlot
Realizes
Dependent 1
Antecedent 1
RealizesExtent
Antecedent 1
CIM_Slot
Description, MaxDataWidth, ThermalRating
CIM_PhysicalComponent
Manufacturer, InstallDate, SerialNumber
Antecedent 1
CIM_PhysicalMemory
Speed, Capacity
CIM_UnitaryComputerSystem
Name, Description, PrimaryOwnerName, PrimaryOwnerContact, ResetCapability
Host 1
Dependent 1
Designates
ComputerSystemPackage
Designation 1
Antecedent 1
ManageWise_NDSName
Label, Tree
ZENworks_SystemInfo
Tag, Model, SerialNumber, Manufacturer, Description, SystemIdentifier, ManagementTechnology



Sample Inventory Database Queries

The following are sample queries for retrieving the inventory information from the ZfS Inventory database.

Refer to the schema diagrams in "Schema Diagrams of CIM and the Extension Schema in ZfS" on page 803 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$,m.label,m.tree FROM managewise.NDSName
m,cim.UnitaryComputerSystem u,managewise.Designates s where
s.Designation=m.id$ and s.HOST=u.id$
```

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

```
SELECT m.Tag,m.Manufacturer,m.SerialNumber FROM
cim.UnitaryComputerSystem u,zenworks.SystemInfo
m,cim.ComputerSystemPackage s WHERE s.Antecedent=m.id$ and
s.Dependent=u.id$
```

3. Retrieve all the software applications with their versions 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 FROM cim.Product m,cim.UnitaryComputerSystem
u,zenworks.InstalledProduct s,managewise.NDSName
m1,managewise.Designates s1 WHERE (s.Product=m.id$ and
s.ComputerSystem=u.id$) AND (s1.Designation=m1.id$ and s1.Host=u.id$) AND
m1.label='SJOHN164_99_139_79.WS' and m1.tree='Novell_AUS'
```

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

```
SELECT
m.DeviceID,m.Family,m.Stepping,m.OtherFamilyDescription,m.MaxClockSpeed,
m.CurrentClockSpeed,m.Role,m.UpgradeMethod FROM cim.Processor
m,cim.UnitaryComputerSystem u,cim.ComputerSystemProcessor s,
managewise.NDSName m1,managewise.Designates s1 WHERE
(s.PartComponent=m.id$ and s.GroupComponent=u.id$) AND
m1.label='SJOHN164_99_139_79.WS'
```

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

```
SELECT name from CIM.UnitaryComputerSystem WHERE
name=’SJOHN164_99_139_79’
```

6. 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
m.DeviceID,m.Family,m.Stepping,m.OtherFamilyDescription,
m.MaxClockSpeed,m.Role,m.UpgradeMethod FROM cim.Processor
m,cim.UnitaryComputerSystem u,
cim.ComputerSystemProcessor s u.id$=? and
s.PartComponent=m.id$ and s.GroupComponent=u.id$
```

   Substitute the ID of the specified inventoried server in place of the ? value for u.id in the query.

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

```
SELECT ip.Address, ipx.Address, mac.MACAddress FROM
cim.IPProtocolEndpoint ip, cim.IPXProtocolEndpoint ipx,
cim.LANEndpoint mac, cim.UnitaryComputerSystem u,
cim.HostedAccessPoint s WHERE (s.Dependent=ip.id$ and
s.Antecedent=u.id$) AND (s.Dependent=ipx.id$ and
s.Antecedent=u.id$) AND (s.Dependent=mac.id$ and
s.Antecedent=u.id$)
```

Modify the same query to get the information for a specified inventoried server as follows:

```
SELECT ip.Address, ipx.Address, mac.MACAddress FROM
cim.IPProtocolEndpoint ip, cim.IPXProtocolEndpoint ipx,
cim.LANEndpoint mac, cim.UnitaryComputerSystem u,
cim.HostedAccessPoint s WHERE (s.Dependent=ip.id$ and
s.Antecedent=u.id$) AND (s.Dependent=ipx.id$ and
s.Antecedent=u.id$) AND (s.Dependent=mac.id$ and
s.Antecedent=u.id$)AND u.id$=?
```

Use the query as shown in query 5 to retrieve the ID of the specified inventoried server and substitute the ID in place of the ? value for u.id in the query.

8. Retrieve the name and other properties of the drives on the hard disk of the specified inventoried server.

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

Use the query shown in query 5to retrieve the ID of the specified inventoried server and substitute the ID in place of the ? for u.id$ in the query.

29 Managing Inventory Information

This section contains the following information to help you customize the way Novell® ZENworks® for Servers (ZfS) displays information:

- "Viewing the Inventory Servers Deployed for Inventory" on page 811
- "Viewing the Inventory Information" on page 812
- "Customizing the Inventory Information" on page 831
- "Exporting the Inventory Data to CSV Format" on page 837

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.

To view the Inventory servers deployed for inventory:

1 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 data between Inventory servers that are situated on different Novell eDirectory trees:

1a In ConsoleOne, select NDS Tree.

1b Click View > Complete Tree View.

1c Select the eDirectory trees or containers within the tree that contains the Inventory servers.

1d Click OK.

2 In ConsoleOne, right-click the Inventory Service object, click View, then click Up Tree View.

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 data 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.

Viewing the Inventory Information

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 data from the last inventory scan for the inventoried server. For more information, see "Viewing the Inventory Summary of an Inventoried Server" on page 813.

- 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 821.

- 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 824.

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 > ZENworks Inventory > 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 data from the last inventory scan for the inventoried server.

To view the inventory information of an inventoried server:

1. Configure the Inventory database.

   For more information, see "Configuring the Inventory Database" on page 812.

2. Right-click an inventoried server, click Actions, then click Inventory.

ZfS provides the following inventory information collected from the inventoried servers:

Scan Data Group	Scan Data Item	Description
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.
	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	Serial number value for the computer, assigned during manufacture
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 > Windows Domain	Windows NT Domain Name	Domain name of the inventoried server
Hardware/Software Inventory > Software > Software Vendors	Name	Name of the software manufacturer
	Version	Version number of the software
	Identification Number	Product ID
	Location	Installation directory
Hardware/Software Inventory > Software > Device Drivers > Pointing Drivers >	Name	Name of the mouse driver
	Version	Version number of the mouse driver

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Software > Device Drivers > Display Drivers	Install Date	Install date of the display driver
	Manufacturer	Name of the display driver manufacturer
	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 servers
	Install Date	Install date of the operating system
	Caption	Operating system name. For example, Windows* 95/ Windows 2000
	Other Description	Page file size
	Role	Type of the operating system, such as server
	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 > Software > NetWare Client Mode	NetWare Client Version	Version number of the NetWare® client
Hardware/Software Inventory > Software > Inventory Scanner Information	Inventory Server	Name of the inventory server to which the scans are sent
	Scan Mode	Mode used by the Scanner to scan the inventoried server
	Version	Version number of the Scanner
Hardware/Software Inventory > Hardware > Pointing Device >	IRQ Number	Interrupt assigned to this device
	Name	Identifying information of the mouse
	Number of Buttons	Number of buttons on the mouse

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Hardware > Keyboard	Delay	Delay before the repeat of a key
	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 > Memory	Total Memory	Total memory of the inventoried servers
Hardware/Software Inventory > Hardware > Display Adapter	Chip Set	Chip set used by the controller to compare stem 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

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Hardware > BIOS	BIOS Identification Bytes	Bytes 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
	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

Scan Data Group	Scan Data Item	Description
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
	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 > 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

Scan Data Group	Scan Data Item	Description
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	Fixed disk drive sectors per track
	Size	Size of the removable disk
Hardware/Software Inventory > Hardware > Disk Drives > Hard Disk > Logical Disk	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 servers
	Description	Description of the CD drive
	Drive Letter	Mapped drive name of the CD drive
	Manufacturer	Vendor Name
	Caption	CD's caption name
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

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Hardware > Bus	Bus Type	Bus type indicates PCI, ISA, and others
	Description	Bus description
	Name	Bus name
	Device ID	Special hexadecimal string identifying the bus type
	Version	Version of the bus supported by the motherboard
Hardware/Software Inventory > Hardware > Network Adapter	Adapter Type	Types of network adapter such as FDDI, token ring, etc.
	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 data 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 inventoried 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

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > Network > 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 > 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 operation
	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

Scan Data Group	Scan Data Item	Description
Hardware/Software Inventory > System > System DMA	Availability	Indicates whether Virtual Direct Memory Access (DMA) is supported
	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 workstation 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 data (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 > ZENworks Inventory > Inventory Query. For more information on how to configure the Inventory database, see "Configuring the Inventory Database" on page 812.

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 for Servers 3 and ZENworks for Desktops 4 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	On (=), 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 > 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 ZfS to start populating the Inventory database with the data you want.

This section covers information on the following sections:

- "Prerequisites for Generating Inventory Reports" on page 824
- "Types of Inventory Reports" on page 825
- "Generating Inventory Reports" on page 827
- "Printing an Inventory Report" on page 829
- "Exporting an Inventory Report to a File" on page 829
- "Running Inventory Reports" on page 824
- "Understanding User-Defined Reports" on page 829

Prerequisites for Generating Inventory Reports

Before running the inventory reports you must make sure that the appropriate ODBC client for Sybase* or Oracle* is installed on the machine running ConsoleOne. The ODBC driver will be automatically configured on the machine when you invoke the Inventory report.

You can install the Sybase ODBC driver version 7.0.0.313 from the *ZENworks for Servers Companion* CD. To install the Sybase ODBC driver, copy the \odbc\sybase\sybaseodbc.zip from the *ZENworks for Servers Companion* CD to a drive. For installation instructions, refer to the odbc\sybase\odbcreadme.txt on the *ZENworks for Servers Companion* CD.

For Oracle, you must install Oracle client only for ODBC because Inventory reports are not compatible with either the older or the later version of the client.

Types of Inventory Reports

You can generate the types of reports described below, assuming you have already configured ZfS to start populating the inventory database with the data 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, Distinguished Name, Distinguished Tree 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, CD ROM, and Network Adapter.	Memory, processor, display details, keyboard, pointing device, fixed and removable disk, floppy, CD drive, and network adapter details for each system
	BIOS Listing	Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, BIOS Install Date, and Manufacturer	List of all the machines with a BIOS manufacturer, BIOS release date, and the total number of such machines
	Devices Listing	Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, and Devices Based on the device selected in the Devices drop-down list, the filter condition for the selected device will be displayed.	List of all machines with a particular device. The devices are pointing device, keyboard, bus, video adapter, network adapter, sound adapter, modem, battery, and power supply.
	Storage Devices Inventory Report	Scope, Distinguished Name, Distinguished Tree Name, IP Address, and DNS Name You can also select to display the following options in the report: Fixed disk and Removable Disk, Logical Disk, Floppy, and CD ROM.	Fixed disk, removable disk, logical disk, floppy, and CD drive details for each system
	Storage Device Listing	Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, and Devices Based on the storage device selected in the Devices drop-down list, the filter condition for the selected device will be displayed.	List of all machines with a particular storage device. The storage devices are fixed and removable disk, floppy, and CD drive
	System Information Listing	Scope, Distinguished Name, Distinguished Tree Name, IP Address, and DNS Name	List of all machines with system information for each machine

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
System Configuration Inventory	Hardware Summary Report	Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, Operating System Type, Operating System Version, Processor Family, Max Clock Speed (Lower Bound in MHz), Max Clock Speed (Upper Bound in MHz), Total Memory (Lower Bound in MB), Total Memory (Upper Bound in MB), Fixed disk Size (Lower Bound in GB), and Fixed Disk Size (Upper Bound in GB)	Operating system name, operating system version, processor family, processor maximum clock speed, memory, and fixed disk size for each machine
	Memory Listing	Show Chart, Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, Total Memory (Lower Bound in MB), and Total Memory (Upper Bound in MB)	List of all the machines within a range of memory size (such as 200-400 MB) and the total number of such machines
	Networking Information Report	Scope, Distinguished Name, Distinguished Tree Name, IP Address, and DNS Name You can also select to display the following options in the report: Network Adapter Type, DNS Name, IP Address, MAC Address, IPX Address, and Windows Domain Name.	Network adapter type, DNS, IP address, MAC address, IPX address, and Windows Domain name for each system
	Operating System Listing	Show Chart, Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, Operating System Type, and Operating System Version	List of all the machines with an operating system type, an operating system version, and the total number of such machines
	Processor Listing	Show Chart, Scope, Distinguished Name, Distinguished Tree 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 machines with a processor family (such as Pentium Pro), processor maximum clock speed, and processor current clock speed of such machines
	System Internal Hardware Inventory Report	Scope, Distinguished Name, Distinguished Tree 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 system
Software Inventory	Application Software Inventory Report	Product Location, Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, Include Product Location, Software Vendor, Software Name, and Software Version	Software with product name, version, vendor, product ID, product location, and recent information for each system
	Software Listing	Include Product Location, Scope, Distinguished Name, Distinguished Tree Name, IP Address, DNS Name, Software Vendor, Software Name, and Software Version	List of all the machines with a software vendor, software name, version, and the total number of such machines

Inventory Report Group	Report Name	Selection Criteria	Information Displayed in the Inventory Report
	Software Summary Listing	Show Chart, Scope, Software Vendor, Software Name, and Software Version	Lists the number of machines with a particular software version **TIP:** The Software Summary Listing chart might not be displayed properly because there is too much software data in your Inventory database. For the chart to be displayed properly, use the selection criteria effectively to restrict the results displayed according to your requirement.
	System Software Inventory Report	Scope, Distinguished Name, Distinguished Tree 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 NetWare Client for each system.
Others	Inventory Scan Listing	Show Chart, Scope, Distinguished Name, Distinguished Tree 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 system
	User Defined Reports For more information on how to create user-defined reports, see "Understanding User-Defined Reports" on page 829	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 > ZENworks Inventory > Configure DB), then click Tools > ZENworks Reports.

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 ZfD 4 and ZfS 3 are installed on the same machine.

For example, if you want to view the inventory information of all inventoried servers, select Server as the scope selection criteria. 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 NT* operating system, you select the Operating System Listing, and specify the selection criteria Scope as Both, the Operating System Type as Windows NT, and the Operating System Version as 3.0.

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 except for Distinguished Name and Distinguished Tree Name. The wildcard characters can be used for character data only.

  For Distinguished Name or Distinguished Tree Name, if you specify only a part of the DN, all machine names containing the specified string in the DN are displayed. For example, if you want to know the list of machine whose Distinguished Name contains novell.invtree, specify novell.invtree as the selection criterion; all the machines whose DN contains novell.invtree are displayed in the Inventory report.

  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.

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 > 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 data 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 or later. For more information on how to create a .RPT file, see the Crystal Report documentation.

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 > Settings > Control Panel > ODBC Data Sources (32 Bit) > Add.

1b Select the ODBC driver for the database you want to connect to.

1c Click Finish.

1d 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.

Customizing the Inventory Information

This section describes how to customize the inventory information.

- "Customizing Software Scanning of Inventoried Servers" on page 831
- "Scanning for Vendor-Specific Asset Information from DMI" on page 833
- "Customizing the Software Scanning Information of Vendors and Products" on page 835
- "Customizing the Hardware Scanning Information of Jaz and Zip Drive Vendors" on page 837

Customizing Software Scanning of Inventoried Servers

You can customize the list of software applications that you want to scan for at the inventoried servers. You specify the software scan settings in the Server Inventory policy page. The software scan settings are saved in eDirectory.

By default, the Scanner will not scan for software applications at the inventoried server. You must enable the Software Scan option in the Server Inventory policy. For more information, see "Configuring the Server Inventory Policy" on page 707.

To specify the applications you want to scan for, you add the list of applications or import files that contain the list of applications. You can also export the list of applications as a file and then modify the file.

If you have a large number of software applications that you want to specify, you can create a Custom Scan file following the conventions explained in this section and later import the file.

To specify software scan settings that you specified at a different location, you export the file at that location and import the file at the location you want to use the list.

The following sections contain more information to help you customize scanning of the inventoried servers:

- "Adding New Applications for Scanning" on page 831
- "Format of the Custom Scan File" on page 832
- "Exporting the List of Application Files for Scanning" on page 833

Adding New Applications for Scanning

To add a new application, you must provide the details of the application.

To add a new application for scanning:

1 In ConsoleOne, open the Server Inventory policy.

For more information, see "Configuring the Server Inventory Policy" on page 707.

Ensure that the Enable Software Scan option is checked.

2. Click the Custom Scan Editor button.
3. Click Add to specify the details of the application.
4. Fill in the details of the application:

   Vendor name, Product name, Product version, Filename, File Size (in Bytes)
5. Click OK.
6. To save the application entry in eDirectory, click OK in the Custom Scan Editor dialog box.

You can also add application entries to the Custom Scan table by importing a file with the list of application entries. You create this file by following the format of the Custom Scan file conventions. For more information, see "Format of the Custom Scan File" on page 832.

To add a list of new applications:

1. Open a text editor.
2. Create a file with the format specified in "Format of the Custom Scan File" on page 832.
3. Save the application as a text file with any extension you prefer.
4. In ConsoleOne, open the Server Inventory policy.

   Ensure that the Enable Software Scan option is checked.
5. Click Custom Scan Editor.
6. Click Import.

   To save the application entry in eDirectory, click OK in the Custom Scan Editor dialog box.

Format of the Custom Scan File

The contents of the Custom Scan file are as follows:

total_number_of_application_entries_in_Custom_Scan_file;
total_number_of_columns_in_the_application_entry

vendor_name;product_name;product_version;file_name;file_size (in Bytes)

vendor_name;product_name;product_version;file_name;file_size (in Bytes)

vendor_name;product_name;product_version;file_name;file_size (in Bytes)

Keep in mind the following guidelines as you work with the Custom Scan file:

- The default total number of columns in the application entry is 5.
- The separator between the columns is a semicolon (;).
- Fill in all the columns for each application entry.
- Do not use comma (,) in the file size parameter.

The following is a sample Custom Scan file:

```
2;5
Novell;GroupWise;5.5;grpwise.exe;4025856
Novell;client32nlm;3.03;client32.nlm;524168
```

Exporting the List of Application Files for Scanning

You can export the Custom Scan file to use at a different location. You export the Custom Scan file at one location and then import it at the other location.

To export the list of applications:

1. In ConsoleOne, open the Server Inventory policy.

   For more information, see "Configuring the Server Inventory Policy" on page 707.

   Ensure that the Enable Software Scan option is checked.
2. Click Custom Scan Editor.
3. Click Export.
4. Type the filename with any extension for the text file.

   The export file is a text file.
5. Click OK.

The exported file will contain the list of applications that are displayed in the Custom Scan table. If you have not saved the list of applications before exporting, the entries in the exported file and the saved application entries in eDirectory will differ.

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 707.
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_pathname_for_asset_tag

DMI1_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag

[SERIALNUMBER]

DMI1_CLASSNAME=DMI_class_pathname_for_serial_number

DMI1_ATTRIBUTEID=DMI_attribute_ID_for_serial_number

[MODEL]

DMI1_CLASSNAME=DMI_class_pathname_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_pathname_for_asset_tag

DMI1_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag

DMI2_CLASSNAME=DMI_class_pathname_for_asset_tag

DMI2_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag

DMI3_CLASSNAME=DMI_class_pathname_for_asset_tag

DMI3_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag

DMI4_CLASSNAME=DMI_class_pathname_for_asset_tag

DMI4_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag

DMI5_CLASSNAME=DMI_class_pathname_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 Software Scanning Information of Vendors and Products

The software information of the same vendor may sometimes have different vendor names or product names. For example, if the software scan data contains information of more than one product for the same vendor, and if the vendor name differs, the inventory display windows will display the software information under different vendor names.

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 product names differ, you can merge the software information.You can also prevent the display of specific vendors and products in the inventory windows. You customize these settings in the Software Rules.

To customize the vendor and product names for display:

1. In the Server Inventory policy, click the Configuration Editor tab.

   For more information, see "Configuring the Server Inventory Policy" on page 707.
2. Click the SWRules suboption, then click Set Defaults.

   The default values are displayed.

```
[vendor]

Novell=Novell Incorporated

Novell Inc=Novell Incorporated

Novell Corporation=Novell Incorporated

Novell Corp=Novell Incorporated

Microsoft=Microsoft Corporation

..

[PRODUCT]

Microsoft® Windows Operating System=NULL

Microsoft ® Windows(TM) Operating System=NULL

Microsoft(R) Windows NT(R) Operating System=NULL

Microsoft(R) Windows (R) 2000 Operating System=NULL

..
```

3. Add or modify the entries.

   The format of SWRules is as follows:

```
[vendor]
```

scanned_vendor_name_reported_by_scanner= vendor_display_name_you_specify

scanned_vendor_name_reported_by_scanner= vendor_display_name_you_specify

```
[product]
```

scanned_product_name_reported_by_scanner= product_display_name_you_specify

scanned_product_name_reported_by_scanner= product_display_name_you_specify

You should follow these rules while editing SWRules:

- Ensure that blank lines do not exist between the sections.
- The section should end with a carriage return.
- Ensure that spaces and symbols in the *scanned_vendor_name_reported_by_the_scanner* and *scanned_product_name_reported_by_the_vendor* do not exist. The scanners compare the *scanned_vendor_name_reported_by_the_scanner* and the *scanned_product_name_reported_by_the_scanner* with the scanned data that they collect. Ensure that names that you use are not case sensitive.

If you specify incorrect entries, the entries preceding the incorrect entry will be used and the other entries will be ignored.

- To modify the vendor name, specify the details for *scanned_vendor_name_reported_by_scanner* and the *vendor_display_name_you_specify*.

  For example, to display the software vendor information for Novell, Novell Inc., Novell Corp, and Novell Inc as Novell Inc., edit the following section:

  ```
  [vendor]
  Novell=Novell Inc.
  NOVELL INC=Novell Inc.
  NOVELL CORP=Novell Inc.
  NOVELL Inc=Novell Inc.
  ```

- To modify the product name, specify the Scanned Product Name and the Product Display Name.

  For example, to display the product information: Novell NetWare (TM) Operating System, Novell NetWare®, Novell NetWare (R) Operating System as Novell NetWare®, edit the following section.

  ```
  [product]
  Novell NetWare (TM) Operating System=Novell NetWare®
  Novell NetWare=Novell NetWare®
  Novell NetWare (R) Operating System=Novell NetWare®
  ```

- To specify that the scanned information for a product or vendor should not be reported by the scanners, add the following entry:

  ```
  [vendor]
  others=null
  ```

4 Click OK.

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 data 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 707.

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

...
```

3 Add or modify the entries.

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.

If you specify incorrect values for the device ID entry, the device will not be displayed in the Inventory windows.

4 Click OK.

Exporting the Inventory Data to CSV Format

You can customize the inventory data you want to export from the ZfS Inventory database in to a comma-separated value (.csv) 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 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:

- "Invoking the Data Export Tool" on page 838
- "Exporting the Inventory Data to a CSV File" on page 838
- "Forming the Query and Setting the Filter Conditions" on page 839
- "Loading an Existing Configuration File" on page 841
- "Running the Data Export Program from the Inventory Server" on page 842

Invoking the Data Export Tool

To invoke the Data Export tool:

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 > ZENworks Inventory > Data Export. For more information on how to configure the Inventory database, see "Configuring the Inventory Database" on page 812.

Exporting the Inventory Data to a CSV File

To export the inventory data to a .CSV file:

1 Open the Data Export tool. See "Invoking the Data Export Tool" on page 838.

2 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 to be included in the query. Based on the inventory components and criteria you specify, the inventory data from the database is exported to a .csv file.

Click Next.

3 Specify the filter conditions for the inventoried servers.

3a Click Edit Query. For more information to how to define a query, see "Forming the Query and Setting the Filter Conditions" on page 839.

3b If you have formed a query with only software attributes (such as Vendor, Name, Version, and Product Identification), the Enable Filter check box will be available for selection.

If you want the results that will be stored in .csv file to be filtered on the basis of the above query, select the Enable Filter check box.

3c Click Next.

4. 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.

   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, 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 files any time you need to.

   5b. Click Next.

6. Select Perform the Query from This Computer to run the data export processing from the workstation computer. This option will access the Inventory database on the specified database server and export the data in to a .csv file.

   If you want to apply default encoding of the machine to the .csv file, select Default Encoding. The Default Encoding check box is selected by default. To apply Unicode encoding to the .csv file, select Unicode Encoding.

7. Specify the .csv filename, then click Finish.

   This generates the .CSV file in the specified directory. Open the .csv file in Microsoft Excel or any other CSV-supported viewer to view the exported data.

8. To run the data export tool from an Inventory server, select the Perform the Query option on a Remote Inventory server. See "Running the Data Export Program from the Inventory Server" on page 842.

9. Save the configuration settings, if necessary.

10. Click Finish.

    If the configuration settings have not been saved, you will be prompted to save the changes.

Forming the Query and Setting the Filter Conditions

To form the query and set the filter conditions for the data export:

1. In ConsoleOne, open the Data Export tool. See "Invoking the Data Export Tool" on page 838.

2. Select Create a New Database Query.

3. Set the scope for exporting the data from the Inventory database.

   If the ConsoleOne snap-ins and the Data Export tool have been installed for both ZENworks for Servers 3 and ZENworks for Desktops 4, the Data Export tool allows you to change the scope of exporting the inventory data.

   By default, the Servers option will be enabled. The query locates all inventoried servers satisfying the query expression. If ZENworks for Servers and ZENworks for Desktops 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.

Also, you must reconfigure the following Database query conditions:

Selecting the Attributes of the Inventory Components: In the Select Attribute window, click the Browse Attribute button to select component attributes. For example, to specify the version of BIOS as a component in the data export, select BIOS as the component, and select Version as the component attribute.

The components are grouped into the following categories: General, Software, Hardware, Network, and System.

The custom attribute will be prefixed by an asterisk (*).

Machines that do not satisfy the query: Select the check box to retrieve machines that do not satisfy the query. By default, this check box is not selected.

Relational operators: The Relational operators determine the relationship between the components and the value. They are grouped on the basis of 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	On (=), 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

For more information on the usage of the relational operators, see "Usage of Relational Operators" on page 824.

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.

Values for the inventory attributes: 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 to the SQL documentation:

Example	Specifies to Include
?	Any one character
_ (underscore)	Any one character
%	Any string of zero or more characters

Example	Specifies to Include
[]	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.

Query connectors and controls: The connectors and controls available for building filter conditions include the following:

AND: The expressions before and after the AND must be true.

OR: Either the expression before the OR or the expression after the OR must be true.

Insert Row: Lets you build the filter condition for this current row.

Delete Row: Deletes the row.

New Group: Lets you form a new filter condition group and specify the criteria for it. This group will be combined with the previous group by using the relational operator specified between the groups.

End: Ends the filter condition.

4. Click OK.

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 file.

To load existing configuration settings for data export:

1. Ensure that you have generated the data configuration files.

   Complete the procedure outlined in "Exporting the Inventory Data to a CSV File" on page 838. This procedure generates the .csv file and the data configuration files.

2. In ConsoleOne, open the Data Export tool. See"Invoking the Data Export Tool" on page 838.

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. Click a saved database query from the list.

   If you want to modify the existing query, click Edit. Otherwise, to proceed with the existing query, click Next.

5 View the data export settings. Click Next.

6 Select the Perform the Query from this Computer option to run the data export processing from the inventoried server. This option will access the Inventory database on the specified database server and export the data in a .csv file.

7 Specify the .CSV filename, then click Finish.

This generates the .CSV file in the specified directory. Open the .csv file in Microsoft Excel or any other CSV-supported viewer to view the exported data.

8 To run the data export tool from an Inventory server, click the Perform the Query on a Remote Server option. See "Running the Data Export Program from the Inventory Server" on page 842.

9 Click Finish.

Running the Data Export Program from the Inventory Server

Running the Data Export program from an Inventory server is recommended if you are exporting data from a large database or if you have specified complex queries with more than 20 database fields selected for exporting.

To run the data export program from the Inventory server:

1 Ensure that you have generated the data configurations files.

Follow Step1 to Step 5 outlined in "Exporting the Inventory Data to a CSV File" on page 838 and ensure that you save the settings in the .exp file.

When you save a .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 Inventory server that has Server Inventory components installed, then click Finish.

3 Copy the .exp file and .cfg file to the Inventory server.

These two files should exist in the same directory on the Inventory server.

From the Inventory server console, run dbexport.ncf on NetWare® servers or dbexport.bat on Windows* NT*/2000 servers, enter
DBEXPORT *"configuration_filename.EXP" "csv_filename.CSV*"

where *configuration_filename.exp* is an existing file that contains the data export settings. The data exported from the database will be stored in the *csv_filename.csv*.

In the above command, you must enter the *configuration_filename.exp* and the *csv_filename.csv* filenames within double quotes.

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 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.

Open the .CSV file in Microsoft Excel or any other CSV-supported viewer to view the exported data.

30 Monitoring Server Inventory Using Status Logs

Novell® ZENworks® for Servers (ZfS) 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 directory on Windows* NT* 4.0/2000 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 directory on Windows NT 4.0/2000 servers.

The Inventory Policy Enforcer writes the status of the current invocation by the policy engine into the invagentpolicyenforcer.log file.

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 c:\invscan directory on Windows NT 4.0/2000 servers. (where C:\ is the root directory if Windows is installed in c:\winnt).

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 data. For example, if your inventory setup consists of a Leaf Server which initiates the roll-up of data 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 scan data from the Inventory server. This log also displays the most recent roll-up time of the scan data 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 scan data 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, 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.
Message Type	Displays the severity of the message.

Status Information	Details
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, click Properties, click Status Report, then click Server Status.

Viewing the Status of the Last Scan on the Inventoried Server

On NetWare, Windows NT 4.0/2000 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 > Tools > ZENworks Inventory > 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, choose the file type, type the filename, then 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 > Tools > ZENworks Inventory > 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 > Status Report > Server Status
Roll-Up Status	Sender, Receiver, Storer	Roll up start time, message	In ConsoleOne, right-click the Inventory Service object, click Properties > Status Report > 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 data:

- 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 NT/2000 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 NT/2000 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 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® for Servers 3 (ZfS). The information will help you to keep current on changes to the documentation.

If you have purchased ZfS 3.0.2 and have not used or installed ZfS 3 or ZfS 3 SP1, you do not need to review this section.

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 documentation updates 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:

March 11, 2005

Updates were made to the following section. The changes are explained below.

- "Setting Up Workstation Inventory" on page 850
- "Understanding the Server Inventory Components" on page 850

Setting Up Workstation Inventory

The following change was made in this section:

Location	Change
"Setting Up the Inventory Database for Oracle8i and Oracle9i" on page 690	Added a new section, "Setting Up the Inventory Database for Oracle9i" on page 693.

Understanding the Server Inventory Components

The following change was made in this section:

Location	Change
"Understanding the Inventory Removal Service" on page 757	Updated the procedure to remove the inventoried servers from the Inventory database.

November 8, 2004

Throughout the guide, graphics have been updated with font changes and newer icons. No notable content changes were made to any graphics.

March 10, 2004

Updates were made to the following sections:

- Setting Up Server Inventory
- Understanding the Server Inventory Components

Setting Up Server Inventory

The following changes were made in this section:

Location	Change
"Setting Up the Inventory Database for MS SQL Server 2000" on page 701	Rewrote the section, "Configuring the Inventory Database for MS SQL Server 2000" on page 701.
"Creating the Inventory Database for Oracle8i on UNIX" on page 691	Specified the Oracle versions (Oracle 8.1.5, 8.1.6, or 8.1.7 Enterprise Edition) supported by the Inventory database on Unix.
"Creating the Inventory Database for Oracle8i on a Windows NT/2000 Server" on page 692	Specified the Oracle versions (Oracle 8.1.5, 8.1.6, or 8.1.7 Enterprise Edition) supported by the Inventory database on a Windows NT/2000 server.
"Configuring the Server Inventory Policy" on page 707	In Step 8f on page 708, explained how to configure the Custom Scan Editor.
"Stopping the Inventory Service" on page 713	Deleted the following the paragraph: To stop all the Inventory services on a Windows NT/2000 Inventory server, at the server console prompt, execute **`stopser "*"`** from *Inventory_server_installation_directory*\inv\server\wminv\bin directory.

Understanding the Server Inventory Components

The following changes were made in this section:

Location	Change
"Understanding the Inventory Service Manager" on page 725 > "Services on Windows NT/2000 Inventory Servers" on page 729	◆ Deleted the following paragraph: To stop all services, enter **`StopSer "*"`** at the command prompt. ◆ Added the procedure to stop the Inventory services on the Windows NT/2000 Inventory server.
"How the Scanners Collect server Inventory Data" on page 731	Updated the information about the Custom Scan Editor.

October 17, 2003

Updates were made to the following sections:

- "Setting Up Server Inventory" on page 853
- "Managing Inventory Information" on page 853

Setting Up Server Inventory

The following change was made in this section:

Location	Change
"Setting Up the Inventory Database" on page 684	Added a new section, "Setting Up the Inventory Database for MS SQL Server 2000" on page 701.

Managing Inventory Information

The following changes were made in this section:

Location	Change
"Generating Inventory Reports" on page 827	Added guidelines to be followed while working with the Reporting dialog.
"Running the Data Export Program from the Inventory Server" on page 842	Updated Step 3 on page 842.

June 27, 2003

Updates were made to the following sections:

- "Setting Up Server Inventory" on page 853
- "Managing Inventory Information" on page 853

Setting Up Server Inventory

The following change was made in this section:

Location	Change
"Possible Inventory Server Configurations for a WAN" on page 676	Added a new deployment scenario, "Scenario 7: Deploying Inventory Server Across Firewall" on page 682

Managing Inventory Information

The following change was made in this section:

Location	Change
"Prerequisites for Generating Inventory Reports" on page 824	Added the following prerequisite for generating Inventory reports from an Oracle database: For Oracle, you must install Oracle client only for ODBC because Inventory reports are not compatible with either the older or the later version of the client.

December 18, 2002

Updates were made to the following sections:

- "Understanding the Server Inventory Components" on page 854
- "Managing Inventory Information" on page 854

Understanding the Server Inventory Components

The following changes were made in this section:

Location	Change
"Understanding the Inventory Scanner" on page 730	Added the following contents in the "Software Information Collected by the Scanners" on page 733 section: If you want to know the mode used by the Scanner to scan the inventoried server, see the inventory summary of the inventoried server > Hardware/Software Inventory > Software > Inventory Scanner Information > Scan Mode. If you want to know the technology available on the inventoried server such as DMI, WMI, and others, see the inventory summary of the inventoried server > Hardware/Software Inventory > General > System Information > Management Technology.

Managing Inventory Information

The following changes were made in this section:

Location	Change
"Running Inventory Reports" on page 824	Added the following note to "Types of Inventory Reports" on page 825 > Software Summary Listing: The Software Summary Listing chart might not be displayed properly because there is too much software data in your Inventory database. For the chart to be displayed properly, use the selection criteria effectively to restrict the results displayed according to your requirement.

September 27, 2002

Updates were made to the following sections:

- Understanding Server Inventory
- Setting Up Server Inventory
- Understanding the Server Inventory Components
- Managing Inventory Information

Understanding Server Inventory

The following change was made in this section:

Location	Change
Chapter 25, "Understanding Server Inventory," on page 653	Renamed the chapter, Introduction to Understanding Server Inventory.

Setting Up Server Inventory

The following changes were made in this section:

Location	Change
"Creating the Inventory Database for Oracle8i on UNIX" on page 691	Added the following prerequisite for configuring the Inventory database: To maintain the Inventory database in Oracle, Server Inventory requires that you have a minimum of twenty five Oracle user licenses.
"Configuring the Server Inventory Policy" on page 707	Added the following para in Step 2 on page 707: Do not select to configure policies in the Solaris or the Linux suboption as they are not supported.
"Starting the Inventory Service" on page 712	Added information on how to start all services on a Windows NT/2000 Inventory server.
"Stopping the Inventory Service" on page 713	Added information on how to stop all services on a Windows NT/2000 Inventory server.

Understanding the Server Inventory Components

The following changes were made in this section:

Location	Change
"Understanding the Inventory Scanner" on page 730	Moved the following sections related to customizing inventory information to "Customizing the Inventory Information" on page 831 in Chapter 29, "Managing Inventory Information," on page 811: ◆ "Customizing Software Scanning of Inventoried Servers" on page 831 ◆ "Scanning for Vendor-Specific Asset Information from DMI" on page 833 ◆ "Customizing the Software Scanning Information of Vendors and Products" on page 835
"Understanding the Inventory Removal Service" on page 757	Removed the Understanding the Server Removal Service section and added a new section to document how to remove the unwanted, redundant, or obsolete inventoried servers from the Inventory database.
"Understanding the Upgrade Service" on page 759	Added a new section to document about the Upgrade service.
"An Overview of the Inventory Components on the Inventory Server" on page 760	Added Upgrade Service to the Server Inventory component table.

Managing Inventory Information

The following changes were made in this section:

Location	Change
"Managing Inventory Information" on page 811	Renamed the chapter, Viewing the Inventory Information to Understanding Server Inventory.
"Viewing the Inventory Information" on page 812	The following updates were made in this section: ◆ Renamed the section, Displaying Inventory Information to Viewing the Inventory Information. ◆ In "Viewing the Inventory Summary of an Inventoried Server" on page 813, updated the table containing the inventory information collected from the inventoried servers. ◆ Updated the entire section, "Viewing Inventory Information of Inventoried Servers by Querying the Database" on page 821. ◆ Updated the entire section, "Running Inventory Reports" on page 824.

Location	Change
"Customizing the Inventory Information" on page 831	Added a new section documenting how to customize the inventory information.
"Exporting the Inventory Data to CSV Format" on page 837	Updated the entire section.

June 18, 2002

Updates were made to the section, Setting Up Server Inventory.

Setting Up Server Inventory

The following changes were made in this section:

Location	Update
"Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool)" on page 686	Updated the entire section.
"Understanding the Sybase Database Startup Parameters" on page 688	Corrected the example for the **-gc** parameter to -gc 120 (was incorrectly documented as -gc).
"Backing Up the Inventory Database Running Sybase" on page 689	Corrected the log filename created by the backup tool.
"Creating the Inventory Database for Oracle8i on a Windows NT/2000 Server" on page 692	Increased the number of minimum user licenses required to maintain the Inventory database in Oracle from five to twenty five.
"Manually Creating the Inventory Database Object for Oracle" on page 695	Step 3c: Changed the value of Database (Write Only) Username to *MWO_UPDATER* (was incorrectly documented as *MWO_WRITER*).

May 17, 2002

Updates were made to the following sections:

- Setting Up Server Inventory
- Understanding Server Inventory
- Monitoring Server Inventory Using Status Logs

Setting Up Server Inventory

The following change was made in this section:

Location	Change
"Changing the Role of the Inventory Server" on page 713	Updated the entire section.

Understanding Server Inventory

The following change was made in this section:

Location	Change
"Understanding ZfS Inventory Attributes" on page 760	Added descriptions to a few ZfS inventory attributes.

Monitoring Server Inventory Using Status Logs

The following change was made in this section:

Location	Change
Chapter 30, "Monitoring Server Inventory Using Status Logs," on page 843	Renamed the chapter, Troubleshooting Server Inventory with Status Logs to Monitoring Server Inventory Using Status Logs.