VI Workstation Inventory

Inventory management involves collecting hardware and software inventory information from managed workstations. This inventory information is scanned and stored in a database, which the network administrator can access to manage the workstations.

Novell® ZENworks® for Desktops (ZfD) lets you gather complete hardware and software inventory information in to a centralized database. The network administrator can query the inventory information of the managed workstations from ConsoleOne®.

The following sections will help you understand and use ZfD Workstation Inventory:

24 Understanding Workstation Inventory

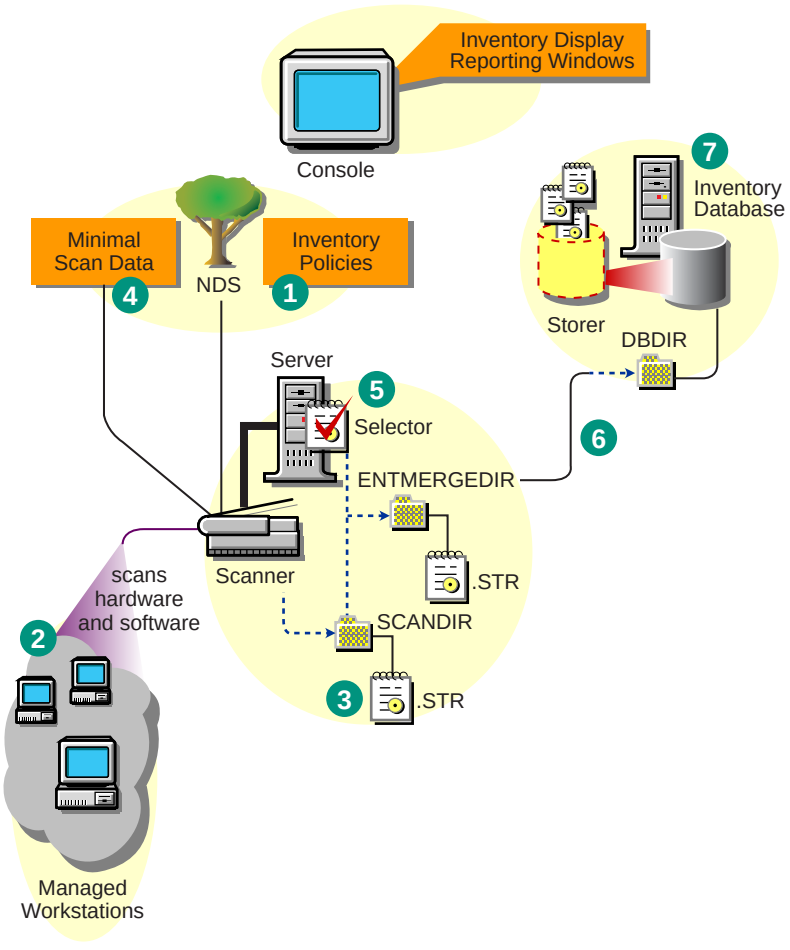
The following sections describe the Novell® ZENworks® for Desktops (ZfD) Workstation Inventory components and processes:

Understanding the Inventory Scanning Cycle

ZfD Workstation Inventory consists of the following components:

- Scanner
- Selector
- Sender
- Receiver
- Storer
- Inventory database

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



The Scanners collect the workstation hardware and software information based on the Inventory policy settings for scheduling and collecting the scan data. This scan information is stored as scan data files (.STR) in the scan directories. A minimal subset of the scan data is also stored in the Novell eDirectory™ Workstation object.

The Selector on the server processes the scan data files to determine whether the new scan data should be merged with the existing data and then places the files in the appropriate directories.

If your inventory deployment requires roll-up of data, the Sender and the Receiver on the servers transfer the compressed scan files from the lower-level servers to the higher-level servers. The Storer on the database server stores the scan data files in the Inventory database. You view the workstation inventory information in ConsoleOne®.

The inventory scanning cycle is as follows:

1. The inventory policies in eDirectory define the inventory settings, such as scanning time, whether to include software scanning of workstations, and the location of the scan directory. These settings are customizable.
2. The Scanner reads the inventory policies and collects the workstation inventory information.
3. The Scanner stores the scan data of each workstation as a .STR file in the scan directory (SCANDIR) at the server.
4. The Scanner stores the minimal inventory information of the workstation in eDirectory.

5. The Selector validates the .STR file and places the file in the enterprise merge directory (ENTMERGEDIR). If there is a database attached, the Selector places the files 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, queries the database, and generates inventory reports in ConsoleOne.

Understanding Rolling Up Scan Data Across Servers

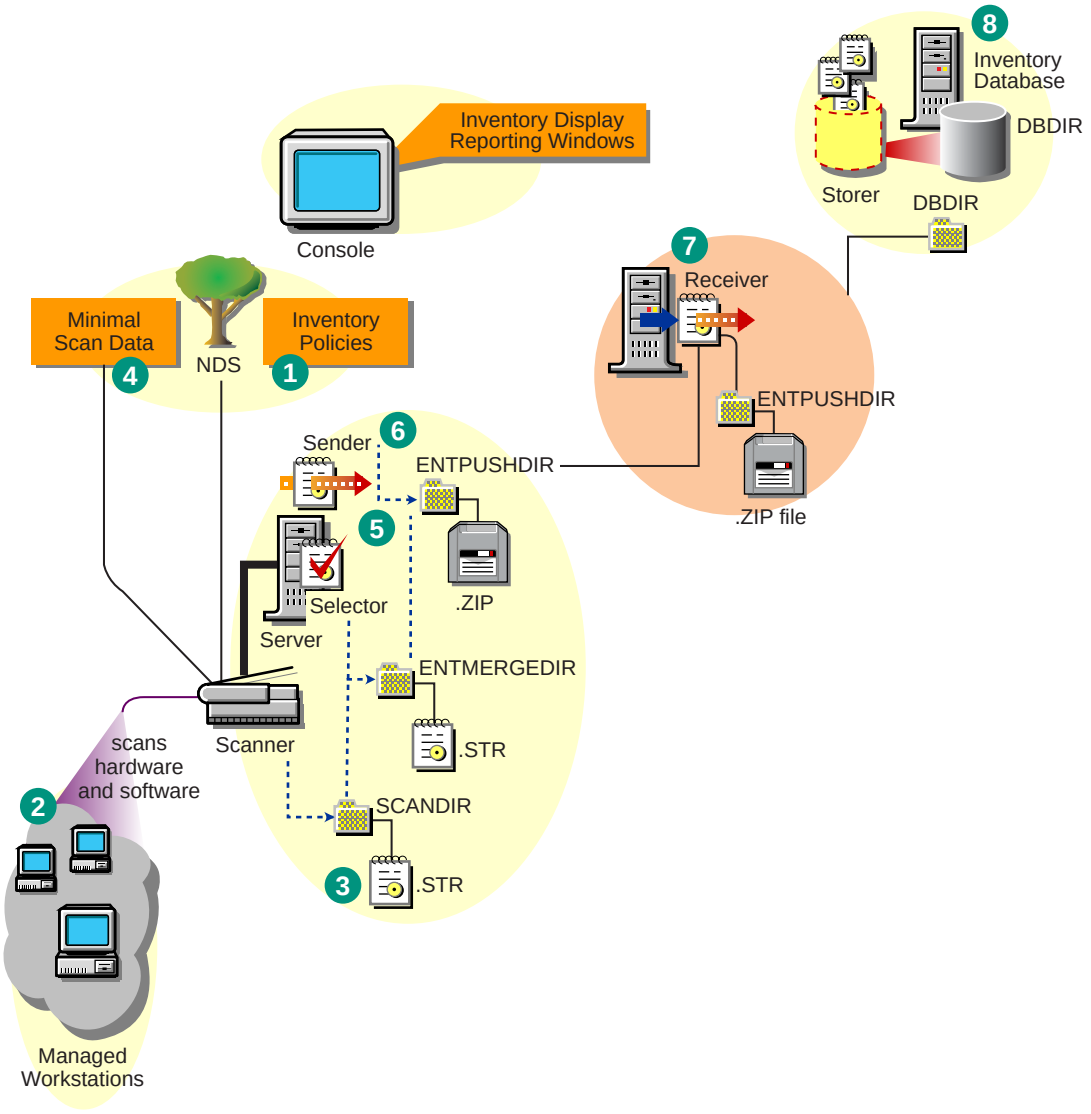
If the inventory deployment rolls up scan data across servers, the process of scanning is as follows:

1. The inventory policies in eDirectory define the inventory settings, such as scanning time, whether to include software scanning of workstations, and the location of the scan directory. These settings are customizable.
2. The Scanner reads the inventory policies and collects the workstation inventory information.
3. The Scanner stores the scan data of each workstation as a scan data file (.STR) in the scan directory (SCANDIR) at the server.
4. The Scanner stores the minimal inventory information of the workstation in eDirectory.
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 places the files in the database directory (DBDIR) also.
6. The Sender on the server has a Roll-Up policy to identify the server to which it will transmit the scan data and the Roll-Up Schedule specifies 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 server.
7. The Receiver on the next-level server receives the .ZIP file.

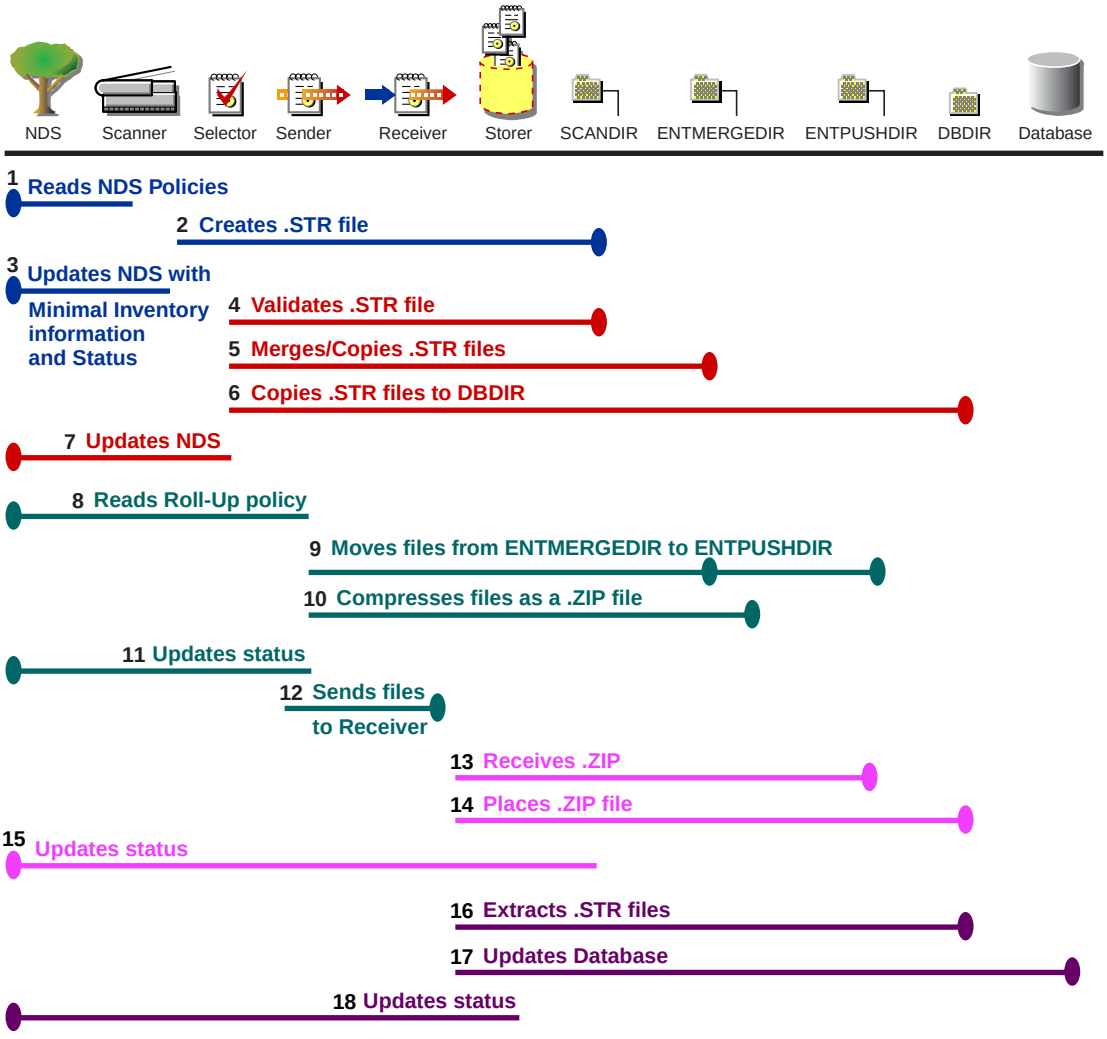
   On the Intermediate Server, the Receiver places the file in the enterprise push directory (ENTPUSHDIR). On the Intermediate Server with Database, or the Intermediate Server with Database and Workstations, the Receiver places the file in ENTPUSHDIR and copies the file in the Database Directory (DBDIR).

   On the Root Server, or the Root Server with Workstations, the Receiver places the file in DBDIR only.
8. The Storer extracts the .ZIP file containing the .STR files in to a temp directory (DBDIR\TEMP) and updates the database with the inventory information of the workstation .STR file.
9. The network administrator views the inventory information, queries the database, and generates inventory reports in ConsoleOne.

The following illustration depicts the scanning process across servers:



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



Understanding the Inventory Scanner

ZfD uses the following platform-dependent scanners to collect workstation hardware and software information:

- WINSCAN.EXE to scan Windows* 95/98 workstations
- NTSCAN32.EXE to scan Windows NT* and Windows 2000 workstations

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

The scan information collected by the scanners is stored as scan data files (.STR) in the scan directories (SCANDIR). A minimal set of scan data is also stored in the eDirectory Workstation object.

The following sections contain detailed information about the Inventory scanners:

- “How the Scanners Collect Workstation Inventory Data” on page 294

Also, see "Customizing the Software Scanning Information of Vendors and Products" on page 380.

How the Scanners Collect Workstation Inventory Data

The scanning process is as follows:

- The Workstation Inventory policy lets you configure the Scheduler to adjust scanning times at the workstations. You can set different actions to run the scanners on one or more managed workstations.
- The Scheduler triggers the Scanner, which reads the following inventory settings from the Workstation Inventory policy and Inventory Service object:
  - **Software Scanning Option:** By default, the Scanner collects the hardware information of the workstations. If the Enable Software Scan option is enabled in the Workstation Inventory policy, the Scanner collects information about software applications.
  - **Custom Scan Editor:** If the Software Scanning option is enabled, the Scanner reports the software information of all .EXE files on the scanned workstations. You configure the applications that you want the Scanner to collect information by using the Custom Scan Editor. For more information, see "Customizing the Software Scanning Information of Vendors and Products" on page 380.
  - **Full Scan:** When scanning the workstation for the first time, the Scanner collects the complete inventory of the workstation, referred to as a *full scan.* After the workstation is scanned, the next time the Scanner compares the current inventory data to the history data that it maintains. If there are any changes to the workstation, the Scanner reports the delta scan data, which contains only the changes in inventory since the last scan was reported. The Delta Scan setting is the default scan operation for each successive scan after the first scanning of the workstation.
  - **Enable Scan of Workstation:** The Scanner collects the inventory information of the workstations associated with the Inventory Service object when this option is enabled.
  - **Location of the SCANDIR Path:** The Scanner stores the scan information of the workstations in the scan data files (.STR) located in the scan directory (SCANDIR) on the inventory server.
  - **Start Full Scan:** The Scanner enforces a complete scanning of the workstation and this setting overrides the option set in the Inventory Service Object property page associated with the workstation.

    For more information, see "Setting Up Workstation Inventory" on page 371.
- The Scanner collects the scan data based on the configurations of the inventory settings.

  If the workstation is instrumented for DMI, the scanners query the DMI Service Layer. For more information, see "DMI-Compliant Scanners" on page 297. If the workstations are WMI

compliant, the scanners also collect the hardware data by querying the WMI information. The scanners also probe the workstations for hardware data.

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

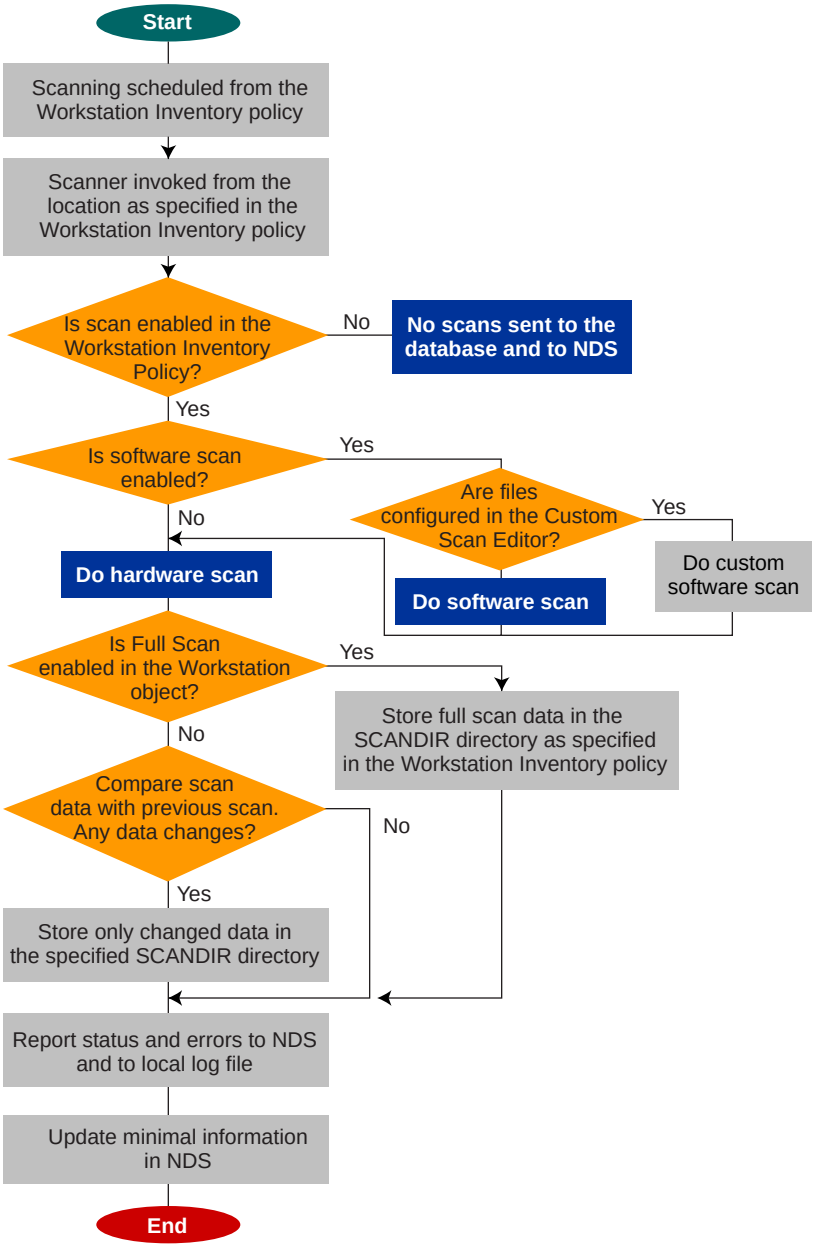
- The scan data of each workstation is stored as .STR files in the SCANDIR directory on the inventory server. The .STR file follows the filename convention: *macaddress_gmt_sequencenumber*.STR, where *macaddress* is the MAC Address of the workstation, *gmt* is the time at which the workstation is scanned for the first time, and *sequencenumber* is the internal sequencing number of the workstation. For example, 00508b12b2c4_944029836000_10.STR is the .STR file for the workstation with the MAC address of 00508b12b2c4, the GMT of 944029836000, and the internal sequencing number of 10.
- The Scanner reports errors in the ZENERRORS.LOG file and updates the status information in the eDirectory Workstation object. The log file is stored in the WINDOWS\TEMP directory on Windows 95/98 workstations or in the TEMP directory on Windows NT/2000 workstations.

  The Scanner tracks the changes in scan data by storing the scan data in the HIST.INI history file. For subsequent scans, the Scanner compares the history scan data with the actual collected data and generates the .STR file.
- The Scanner stores minimal information of the scan data in the MINFO.INI file located in WINDOWS\TEMP on Windows 95/98 workstations and in TEMP on Windows NT/2000 workstations.

  The Scanner updates the eDirectory object with the scan data contained in MINFO.INI.

Scanning Process Flow Chart

The following flow chart illustrates the hardware and software scanning process:



Summary of Files that the Scanner Processes

A summary of the files that the Scanner processes follows:

Filename	Description	Location
filename.STR	Contains the scan data of each workstation.	Scan directory (SCANDIR).
HIST.INI	Contains the history of scan data for each workstation.	WINDOWS\TEMP on Windows 95/98 workstations, or TEMP on Windows NT/2000 workstations.
ZENERRORS.LOG	Contains the error and status information for the latest scan of the workstation.	WINDOWS\TEMP on Windows 95/98 workstations, or TEMP on Windows NT/2000 workstations.

Software Information Collected by the Scanners

The scanners follow this process for software scanning:

- Collects the information about the software on the workstations.
- Customizes the software scanning using the Custom Scan Editor.

  By default, the software scanning includes collecting information of files with .EXE file extensions. See "Customizing the Software Scanning Information of Vendors and Products" on page 380 for information on the list of software applications you can scan for.

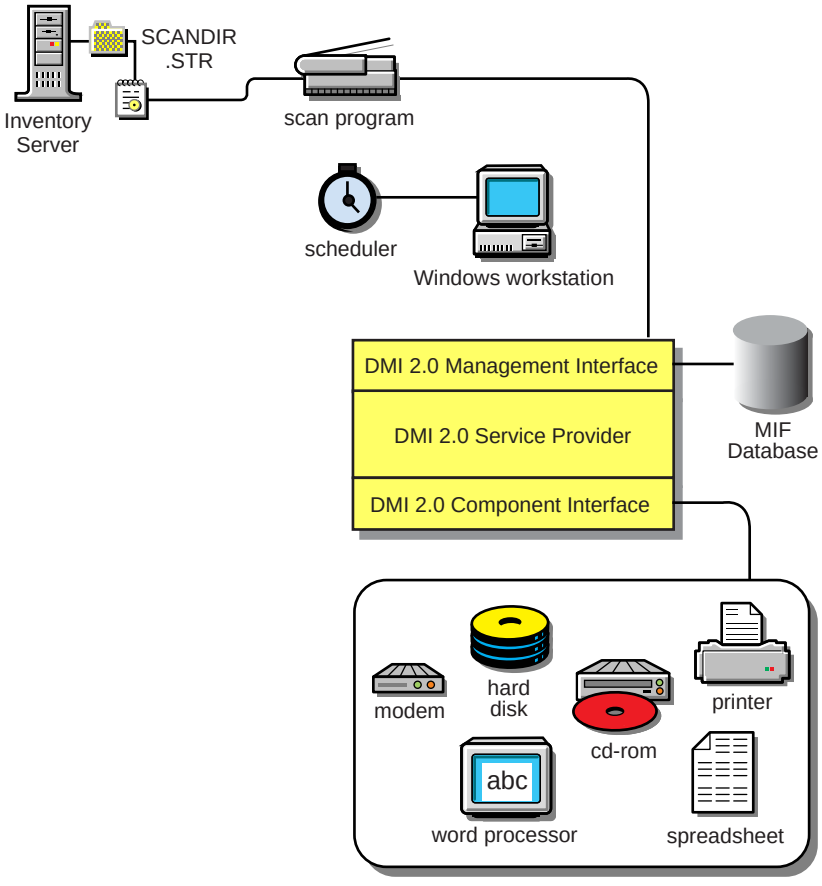
  If the software applications on the workstation are installed using Microsoft* Installer, the scanners use the information from Microsoft Installer (MSI). Otherwise, the scanners collect the software information from the header of the software application files.
- Reports the information about the scanned 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, query, or generate reports of the software information.

DMI-Compliant Scanners

The scanners for scanning workstations (Windows 95/98 and 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 workstation. The scanners will scan for specific components that are instrumented on the workstation 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 managed workstations, you need to instrument the workstation by installing the vendor-specific components. To install the DMI 2.0 Service Layer on the workstations, use the DMISLSnappshot.AOT file located in the \PUBLIC\ZENWORKS\DMISNAPSHOT directory to create a NAL Application Object.

HINT: If the workstations are DMI compliant, the scanners will collect hardware data by querying the DMI Service Layer. Otherwise, the scanners probe the workstations.

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

For example, if you have a Compaq* Family Deskpro EN Model-SFF6500 workstation running under Windows 98, download the Management Product software - Compaq Insight Management Desktop Agents software for Windows 95/98/NT from the Compaq Web site.

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

WMI-Compliant Scanners

The scanners collect hardware data from Windows 95/98/NT/2000 workstations based on Microsoft's Windows Management Instrumentation (WMI) specification.

WMI is the Microsoft implementation of Web-Based Enterprise Management (WBEM) that enables accessing management information in an enterprise environment. WMI 1.5 is fully compliant with Common Information Model (CIM) schema, which is an industry standard. For

more information, see Microsoft WMI Web site (http://www.microsoft.com/hwdev/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 workstation. The scanners also scan for specific components that are instrumented on the workstation through WMI.

WMI-compliant scanners are supported on Windows 98, Windows 95, and WIndows NT/2000 managed workstations only.

You can view the WMI data of the managed workstations in the Workstation Inventory Summary.

To obtain WMI information from the workstation:

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

   To instrument a workstation for WMI, you need to download only the WMI Core Software Installation.

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

   To troubleshoot any WMI-related problems, you can avail WMI SDK download.

2. Install WMI Core Software on Windows NT/95/98 workstations.

3. On the server, modify the \PUBLIC\ZENWORKS\SCANSOURCE.INI file. Update the following entry in this file:

   ```
   [SOURCE]

   do WMIScan= scansetting
   ```

   where *scansetting* is set to either *TRUE or FALSE*.

   By default, scansetting is set to TRUE and WMI scanning is enabled. To disable WMI scanning, set this to FALSE.

   The scansetting modifications effect only those workstations that are connected to the server. For example, if you set the scansetting to False on a server, such as AUS-INV-SERVER, only those workstations that are connected to this server will not be scanned for WMI.

Hardware Data Collected by the Scanners

The scanners collect the following hardware information.

The following table contains the DMI/WMI components that are addressed in the DMI/WMI information.

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.AssetTag	Win32_SystemEnclosure.SMBIOSAssetTag
System.Model	Not applicable	Win32_SystemEnclosure.Model
System.ModelNumber	Not applicable	Win32_SystemEnclosure.SerialNumber

Scan Data	DMI Class and Attribute	WMI Class and Attribute
System.DNName	Not applicable	Not applicable
System.TreeName	Not applicable	Not applicable
IPX.Address	Not applicable	Win32_NetworkAdapterConfiguration.IPX Address (Only on Windows NT/2000)
IP.Address	Not applicable	Win32_NetworkAdapterConfiguration.IPAddress (Only on Windows NT/2000)
IP.Subnet (Subnet Mask)	Not applicable	Win32_NetworkAdapterConfiguration IPSubnet (Only on Windows NT/2000)
DNS.HostName	Not applicable	Win32_NetworkAdapterConfiguration.DNS HostName + Win32_NetworkAdapterConfiguration.DNS Domain (Only on Windows NT/2000)
NetworkAdapter.MACAddress	Not applicable	Win32_NetworkAdapterConfiguration.MAC Address (Only on Windows NT/2000)
Modem.Description	Not applicable	Win32_POTSModem.Description
Modem.Name	Not applicable	Win32_POTSModem.Name
Modem.Vendor	Not applicable	Win32_POTSModem.ProviderName
NetworkAdapter.DriverDescription	DMTF|Network Adapter Driver |001.Driver Software Name	Win32_SystemDriver.Description (Only on Windows NT/2000)
NetworkAdapter.DriverName	DMTF|Network Adapter Driver |001.Driver Software Name	Win32_SystemDriver.PathName (Only on Windows NT/2000)
NetworkAdapter.DriverVersion	DMTF|Network Adapter Driver |001.Driver Software Version	Not applicable
NetworkAdapter.Speed	DMTF|Network Adapter 802 Port|001.5	Win32_NetworkAdapter.MaxSpeed (Only on Windows NT/2000)
NetworkAdapter.Name	DMTF|Network Adapter Driver|001.Driver Software Description	Win32_NetworkAdapter.Name (Only on Windows NT/2000)
NetworkAdapter.PermAddress	DMTF|Network Adapter 802 Port|001.PermanentNetworkAddress	Win32_NetworkAdapter.PermanentAddress (Only on Windows NT/2000)
Login.LoginName	Not applicable	Not applicable
Login.DomainName	Not applicable	Win32_ComputerSystem.Domain (Only on Windows NT/2000)
NWClient.Version	Not applicable	Not applicable
Processor.stepping	DMTF|Processor|004.ProcessorVersionInformation	Win32_Processor.Stepping
Processor.DeviceID	Not applicable	Win32_Processor.DeviceID
Processor.Family	DMTF|Processor|004.Processor.Family	Win32_Processor.Family

Scan Data	DMI Class and Attribute	WMI Class and Attribute
Processor.MaxClockSpeed	DMTF|Processor|004.MaximumSpeed	Win32_Processor.MaxClockSpeed
Processor.CurrentClockSpeed	DMTF|Processor|004.CurrenteSpeed	Win32_Processor.CurrentClockSpeed
Processor.Role	DMTF|Processor|004.ProcessorType	Win32_Processor.ProcessorType
Processor.Upgrade	DMTF|Processor|004.ProcessorUpgrade	Win32_Processor.UpgradeMethod
Processor.OtherFamily	Not applicable	Win32_Processor.OtherFamilyDescription
BIOS.Manufacturer	DMTF|SystemBIOS|001.BIOSManufacturer	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.BIOSVersion	Win32_BIOS.SMBIOSBIOSVersion
BIOS.PrimaryBIOS	DMTF|SystemBIOS|001.BIOSPrimaryBIOS	Win32_BIOS.PrimaryBIOS
BIOS.Size	DMTF|SystemBIOS|001.BIOSROMSize	Not applicable
Bus.Version	Not applicable	Not applicable
Bus.Type	DMTF|Bus Port|001.Protocol	Not applicable
Mouse.DeviceType	Not applicable	Not applicable
Mouse.Type	DMTF|Mouse|004.Mouse Interface	Not applicable
Mouse.NumberOfButtons	DMTF|Mouse|004.Mouse Buttons	Not applicable
Mouse.DriverName	DMTF|Mouse|004.Mouse Driver Name	Not applicable
Mouse.DriverVersion	DMTF|Mouse|004.Mouse Driver Version	Not applicable
Mouse.IRQ	DMTF|Mouse|004.Mouse IRQ	Not applicable
Keyboard.Layout	DMTF|Keyboard|003.Layout	Win32_Keyboard.Layout
Keyboard.Subtype	Not applicable	Not applicable
Keyboard.Type	DMTF|Keyboard|003.Keyboard.Type	Win32_Keyboard.Description
Keyboard.Fkeys	Not applicable	Win32_Keyboard.NumberOfFunctionKeys
Keyboard.Delay	Not applicable	Not applicable
Keyboard.TypematicRate	Not applicable	Not applicable
Monitor.NumberOfColorPlanes (NEW)	Not applicable	Win32_PCVideoController.NumberOfColor Panes
Monitor.HorizontalResolution	DMTF|Video|004.Current Horizontal Resolution	Win32_PCVideoController.CurrentHorizont alResolution

Scan Data	DMI Class and Attribute	WMI Class and Attribute
Monitor.VerticalResolution	DMTF|Video|004.Current Vertical Resolution	Win32_PCVideoController.CurrentVertical Resolution
Monitor.DisplayType	DMTF|Video|004.Video Type	Win32_PCVideoController.VideoArchitectu re
Monitor.MemoryType	DMTF|Video|004.Video Memory Type	Win32_PCVideoController.VideoMemoryT ype
Monitor.MaxMemorySupported	DMTF|Video|004.Video RAM Memory Size	Win32_PCVideoController.MaxMemorySu pported
Monitor.Bitsperpixel	DMTF|Video|004.Current Number of Bits per Pixel	Win32_PCVideoController.CurrentBitsPer Pixel
Monitor.ControllerDescription	DMTF|Video|004.Video Controller Description	Win32_PCVideoController.Description
Monitor.MaxRefreshrate	DMTF|Video|004.Maximum Refresh Rate	Win32_PCVideoController.MaxRefreshRat e
Monitor.MinRefreshrate	DMTF|Video|004.Minimum Refresh Rate	Win32_PCVideoController.MinRefreshRat e
Monitor.VideoBIOSManufacturer	DMTF|Video BIOS|001.BIOS Manufacturer	Not applicable
Monitor.VideoBIOSVersion	DMTF|Video BIOS|001.Video.BIOS Version	Not applicable
Monitor.VideoBIOSReleaseDate	DMTF|Video BIOS|001.Video.BIOS Release Date	Not applicable
Monitor.VideoBIOS.IsShadowed	DMTF|Video BIOS|001.Video.Shadowing State	Not applicable
ParallelPort.Name	DMTF|Parallel Ports|003.Logical Name	Not applicable
ParallelPort.Address	DMTF|Parallel Ports|003.Parallel Base I/O Address	Not applicable
ParallelPort.DMASupport	DMTF|Parallel Ports|003.DMA Support	Not applicable
ParallelPort.IRQ	DMTF|Parallel Ports|003.IRQ Used	Not applicable
SerialPort.Name	DMTF|Serial Ports|004.Logical Name Pointing Device Port Name	Not applicable
SerialPort.Address	DMTF|Serial Ports|004.Serial Base I/O Address	Not applicable
SerialPort.IRQ	DMTF|Serial Ports|004.IRQ Used	Not applicable
FloppyDrive.DriverName	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 = [2,11]or[13,22])
FloppyDrive.Vendor	Not applicable	Not applicable

Scan Data	DMI Class and Attribute	WMI Class and Attribute
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 = [2,11]or[13,22])
FloppyDrive.MaxNumberofCylinders	DMTF|Disks|003.Number of Physical Cylinders	Not applicable
FloppyDrive.NumberOfHeads	DMTF|Disks|003.Number of Physical Heads	Not applicable
FloppyDrive.Sectors	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 = [2,11]or[13,22])
CDROMDrive.Name	DMTF|Logical Drives|001.Logical Drive Name	Win32_CDROMDrive.Drive
CDROMDrive.Vendor	Not applicable	Win32_CDROMDrive.Manufacturer
CDROMDrive.Description	DMTF|Disks|003.Interface.Description (when DMTF|Disks|003.Storage Type=Compact Disk Disk(8))	Win32_CDROMDrive.Description
CDROMDrive.DeviceID	DMTF|Disks|003.DeviceID	Win32_CDROMDrive.Caption
HardDrive.Vendor	Not applicable	Win32_DiskDrive.Manufacturer (Win32_DiskDrive.MediaType = "Fixed Hard Disk media")
HardDisk.Description	DMTF|Disks|003.Interface Description (when DMTF|Disks|003.Storage Type=Hard Disk(3))	Win32_DiskDrive.Description (Win32_DiskDrive.MediaType = "Fixed Hard Disk media")
HardDisk.Cylinders	DMTF|Disks|003.Number of Physical Cylinders	Win32_DiskDrive.TotalCylinders (Win32_DiskDrive.MediaType = "Fixed Hard Disk media")
HardDisk.Heads	DMTF|Disks|003.Number of Physical Heads	Win32_DiskDrive.TotalHeads (Win32_DiskDrive.MediaType = "Fixed Hard Disk media")
HardDisk.Sectors	DMTF|Disks|003.Number of Physical Sectors per Track	Win32_DiskDrive.SectorsPerTrack (Win32_DiskDrive.MediaType = "Fixed Hard Disk media")

Scan Data	DMI Class and Attribute	WMI Class and Attribute
HardDisk.Capacity	DMTF|Disks|003.Total Physical Size	Win32_DiskDrive.Size (Win32_DiskDrive.MediaType = "Fixed Hard Disk media")
HardDisk.PartitionName	DMTF|Partition|002.Partition Name	Not applicable
HardDisk.PartitionSize	DMTF|Partition|002.Partition Size	Not applicable
HardDisk.PartitionFileSystemType	DMTF|Partition|002.FileSystem	Not applicable
LogicalDrive.Name	DMTF|Logical Drives|001.Logical Drive Name (when DMTF|Logical Drives|001.Logical Drive Type=Fixed Drive(3))	Win32_LogicalDiskDeviceID (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
LogicalDrive.Size	DMTF|Logical Drives|001.Logical Drive Size	Win32_LogicalDisk.Size (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
LogicalDrive.FreeSpace	Not applicable	Wind32_LogicalDisk.FreeSpace (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
LogicalDrive.VolumeSerialNumber	Not applicable	Win32_LogicalDisk.VolumeSerialNumber (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
LogicalDrive.Volume (Volume Label)	Not applicable	Win32_LogicalDisk.VolumeName (when Win32_LogicalDisk.DriveType = 3 (Local Disk))
Operating System.Name	DMTF|Operating System|001.Operating System Name	Win32_OperatingSystem.OSType
OperatingSystem.Version	DMTF|Operating System|001.Operating System Version	Win32_OperatingSystem.Version
OperatingSystem.Codepage	Not applicable	Win32_OperatingSystem.CodeSet
OperatingSystem.InstallDate	Not applicable	Win32_OperatingSystem.InstallDate
OperatingSystem.TotalSwapSpaceSize	DMTF|System Memory Settings|001.Total Size of Paging Files	Win32_OperatingSystem.SizeStoredInPagingFiles
OperatingSystem.Description	DMTF|Operating System|001.Operating System Description	Win32_OperatingSystem.Caption
VirtualMemory.TotalVirtualMemorySize	DMTF|System Memory Setting|001.Total Virtual Memory	Not applicable
InventoryScanner.Version	Not applicable	Not applicable
InventoryScanner.LastScanDate	Not applicable	Not applicable

Scan Data	DMI Class and Attribute	WMI Class and Attribute
InventoryScanner.InventoryServer	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
Scsidrive.Description	Not applicable	Win32_DiskDrive.Description (Win32_DiskDrive.MediaType = "Removable media")
Scsidrive.Manufacturer	Not applicable	Win32_DiskDrive.Manufacture (Win32_DiskDrive.MediaType = "Removable media")
Scsidrive.Name (mapped drive)	Not applicable	Not applicable
Memory.Size	DMTF|System Memory Settings|001.Total Physical Memory	Not applicable
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
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.Verify	Not applicable	Not applicable
MotherBoard.BoardRev	Not applicable	Not applicable
MotherBoard.Slots	DMTF|Motherboard|001.Number of Expansion slots	Not applicable
Battery.Name	DMTF|Portable Battery|002.Portable Battery Device Name	Win32_Battery.Name

Scan Data	DMI Class and Attribute	WMI Class and Attribute
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 Smart Battery Version	Win32_Battery.SmartBatteryVersion
Battery.Manufacturer	DMTF|Portable Battery|002.Portable Battery Manufacturer	Win32_PortableBattery.Manufacturer
Battery.ManufacturerDate	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 Suppy Input Voltage Capability Description	Win32_UninterruptiblePowerSupply.Description
PowerSupply.Power	DMTF|Power Supply|002./Total Output Power	Win32_UninterruptiblePowerSupply.TotalOutputPower
DMA.Number	DMTF|DMA|001.DMA Number	Win32_DMA.DMAChannel
DMA.Description	DMTF|DMA|001.DMA Description	Win32_DMA.Description
DMA.Availability	DMTF|DMA|001.DMA Channel Availability	Win32_DMA.Availability
DMA_BurstMode	DMTF|DMA|001.DMA BurstMode	Win32_DMA.BurstMode
UCS.PrimaryOwnerContact	DMTF|General Information|001.3	Win32_UnitaryComputerSystem.PrimaryOwnerContact
UCS.PrimaryOwnerName	DMTF|General Information|001.4	Win32_UnitaryComputerSystem.PrimaryOwnerName
IRQ.Number	DMTF|IRQ|002.IRQNumber	Win32_IRQ.IRQNumber
IRQ.Availability	DMTF|IRQ|002.Availability	Win32_IRQ.Availability
IRQ.TriggerType	DMTF|IRQ|002.TriggerType	Win32_IRQ.TriggerType
IRQ.Shareable	DMTF|IRQ|002.Shareable	Win32_IRQ.Shareable
CIM_Card.Description	DMTF|System Slots|003.Description	Not applicable
CIM_Slot.MaxDataWidth	DMTF|System Slots|003.MaxDataWidth	Not applicable
CIM_Slot.ThermalRating	DMTF|System Slots|003.Tr	Not applicable

NOTE: PCMCIA modems are connected to the computer through the PCMCIA slots on the workstations. The Scanner detects PCMCIA modems that are active on the computer. If you want to know which modem is installed on the computer, use the Windows System Device Manager on the Windows workstation.

Non-PCMCIA modems are connected to the computer through the external ports. For example, some non-PCMCIA modems are connected through the serial ports. The Scanner detects non-PCMCIA modems that are installed on the computer.

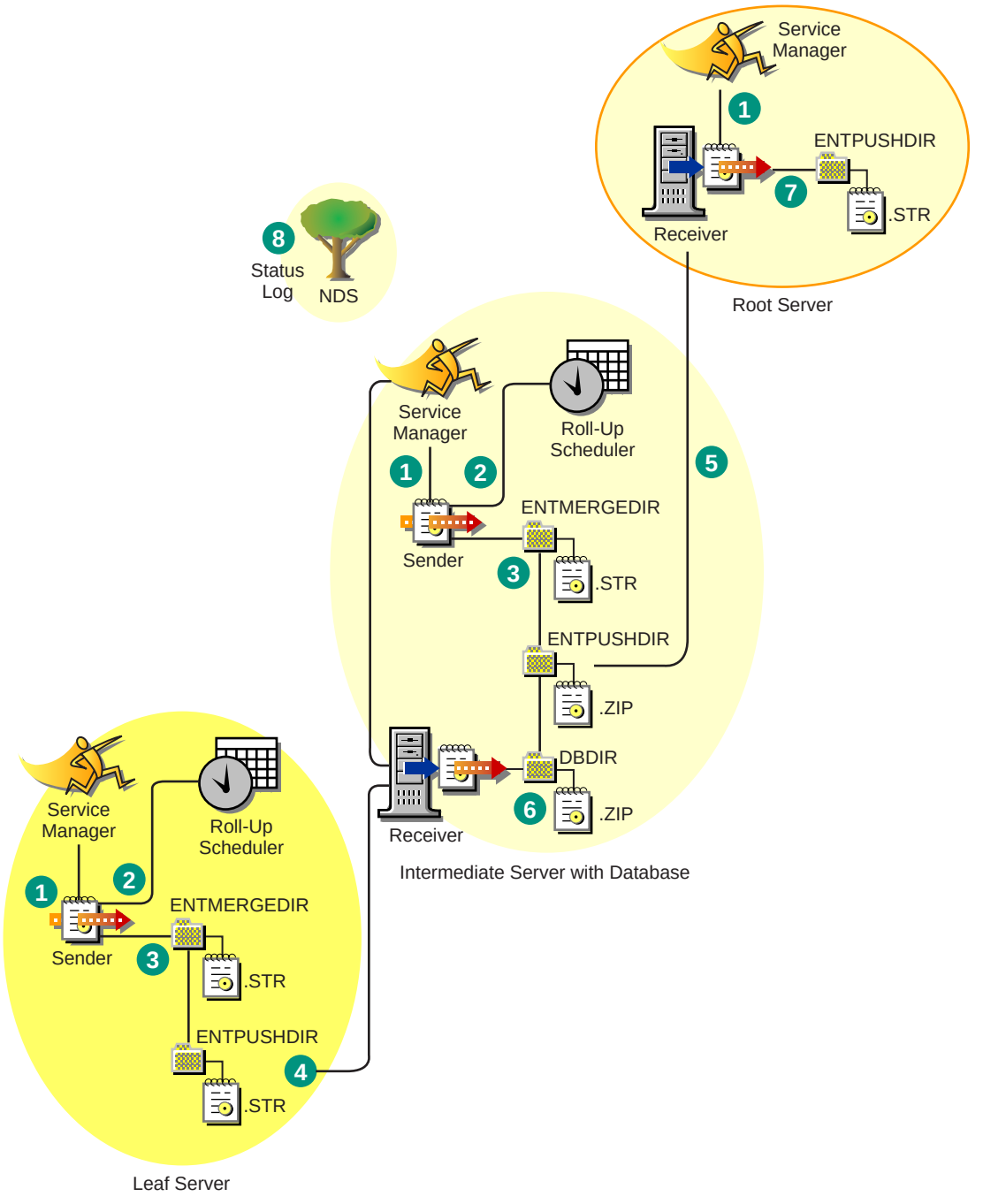
Non-PCMCIA modems may not be active at the time of scanning. Also, these modems may not be connected, though configured on the computer. In this case, the Scanner detects the modem and reports the scan information of the modem.

Understanding the Sender-Receiver

The Sender and the Receiver on the servers transfer the scan files from the lower-level servers to the higher-level servers. The following sections contain more information:

- "Understanding the Sender" on page 309
- "Understanding the Receiver" on page 309
- "Understanding the Compressed Scan Data File" on page 310
- "Sender-Receiver Directories" on page 310

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. The Sender sends the .ZIP file from the ENTPUSHDIR directory to the Receiver on the next-level server.

5. The Sender on the Intermediate Server sends the .ZIP file to the Receiver on the next-level server.
6. The Receiver copies the .ZIP files to the ENTPUSHDIR directory.
7. The Receiver copies the .ZIP files to the database directory (DBDIR).
8. 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 "Inventory Components on Servers" on page 322 for a quick reference table of 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 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 server as a .ZIP file and then deletes the .STR files. For more information, see "Understanding the Compressed Scan Data File" on page 310.
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 receives an acknowledgment from the Receiver that a .ZIP file was properly received and then deletes the compressed files in the ENTPUSHDIR directory.
6. After the roll-up of data, the Sender updates the zeninvRollUpLog attribute of the server on which the compressed file was created with the following details: 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 server to which it was sent.

   The status information for all actions of the Sender is logged in the Roll-Up Log and Server Status log. For more information, see "Troubleshooting Workstation Inventory with Status Logs" on page 431.

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 "Inventory Components on Servers" on page 322 for a quick reference table of server components.

The processing done by the Receiver is as follows:

1. On successfully establishing communication with the Sender, the Receiver receives the scan .ZIP file from the Sender. The file is placed in the enterprise push directory (ENTPUSHDIR).

   On an Intermediate Server, the file is placed in ENTPUSHDIR. On an Intermediate Server with Database, or an Intermediate Server with Database and Workstations, the file is placed in ENTPUSHDIR and copied to the Database Directory (DBDIR).

2. The Receiver on the Root Server or the Root Server with Workstations receives the .ZIP files from the Senders and copies the files to the DBDIR directory on the server.

3. The Receiver logs the status information in the Roll-Up log. For more information, see "Troubleshooting Workstation Inventory with Status Logs" on page 431.

Understanding the Compressed Scan Data File

The Sender compresses the scan data files (.STR) into a .ZIP file. The .ZIP file is named using the following naming conventions:

scheduledtime_inventoryservername_siteID_sitename.ZIP

where *scheduledtime* refers to the date and time when the Sender is scheduled for roll-up of scan information, *inventoryservername* refers to the inventory server on which the .ZIP file was compressed, *siteID* refers to the identification of the database that is attached to the inventory server, *sitename* refers to the unique site name of the database specified during installation, and *ZIP* is the file extension for the compressed files.

The .ZIP filename changes depending on if the database is attached to the server. If the database is not attached to the server, the file is named as follows:

*scheduledtime_inventoryservername*_NOTSTOREDINDATABASETILLNOW_NULL.ZIP

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

scheduled_time_inventoryservername.PRP

The property file identifies the information for roll-up from the enterprise push directory (ENTPUSHDIR) to the next-level server. 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	DBDIR
Leaf Server, Leaf Server with Database	Runs on this server	--	• Sender moves the STR files to the ENTPUSHDIR.	• Sender compresses the .STR files as a .ZIP file. • Sender deletes the .STR files. • Sends the .ZIP file to the next-level server.	--

Server	Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
Intermediate Server	Runs on this server	Runs on this server	--	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender sends the .ZIP files to the next-level server.	--
Intermediate Server with Workstations	Runs on this server	Runs on this server	◆ Sender moves the .STR files to the ENTPUSHDIR.	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender compresses the .STR files in to .ZIP files. ◆ Sender deletes the .STR files. ◆ Sender sends the .ZIP files to the next-level server.	
Intermediate Server with Database	Runs on this server	Runs on this server	--	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender sends the .ZIP file to the next-level server.	◆ Receiver copies the file in this directory.
Intermediate Server with Database and Workstations	Runs on this server	Runs on this server	◆ Sender moves the .STR files to the ENTPUSHDIR.	◆ Receiver receives the .ZIP files from the lower-level server in 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 server.	◆ Receiver copies the file in this directory.
Root Server, Root Server with Workstations	--	Runs on this server	--	--	◆ Receiver receives the .ZIP files from the lower-level server in this directory.

On the Standalone Server, the Receiver is not loaded.

Understanding the Selector

The Selector is a Java component on the server that receives the scan data from the workstations. These servers can be any of the following: Leaf Server, Leaf Server with Database, Intermediate Server with Database and Workstations, Intermediate Server with Workstations, Root Server with Workstations, and Standalone Server. See "Inventory Components on Servers" on page 322 for a quick reference table of server components.

The processing done by the Selector is as follows:

1. While scanning the workstation, the Scanner creates a scan data file (.STR) in the scan directory (SCANDIR) at the server for each scan done on the workstation. The location of SCANDIR is obtained from the Inventory Service object. The Selector processes the .STR files placed by the Scanner in the SCANDIR directory.
2. The Selector checks the validity of the .STR file to ensure that it is a valid file generated by the Scanner. The Selector processes only valid .STR files. If invalid files are present in the directory, the Selector deletes them.
3. Based on the role of the server, the Selector copies the .STR files to the appropriate directories:
   - If a full scan is done at the workstation, the Selector on the server copies the .STR file 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. See the table below.
   - If a delta scan is done at the workstation, the Selector on the server appends the contents of the file in to the existing files in the DBDIR directory (if the database is attached) and the ENTMERGE directory.

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

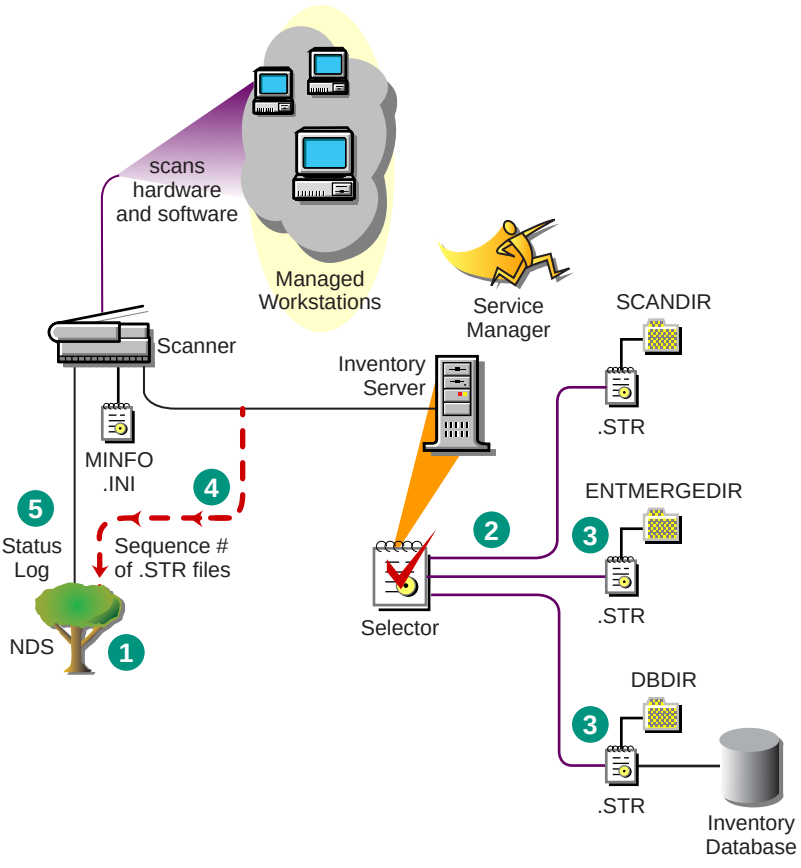
Server	Copies the .STR file to the Database Directory (DBDIR)	Copies the .STR file to the Enterprise Merge Directory (EntMergeDir)
Leaf Server with Database	Yes	Yes
Leaf Server	--	Yes
Intermediate Server with Database & Workstations	Yes	Yes
Standalone Server	Yes	--
Root Server with Workstations	Yes	--

4. The Selector updates the sequence number of valid .STR files in eDirectory.

   The Selector determines whether to enforce a full scan if the scanning sequence number in the .STR file is improper. If the .STR file is invalid or if there are discrepancies in the sequence number of the .STR file, the Selector enforces a full scan.
5. The Selector logs the status in the Server log. For more information, see "Troubleshooting Workstation Inventory with Status Logs" on page 431.

   The Selector removes the existing .STR files in the SCANDIR directory.

The following illustration depicts the processing done by the Selector:



Understanding the Storer

The Storer is a Java component on the server that has a database attached to it. These servers can be any of the following: Leaf Server with Database, Intermediate Server with Database, Intermediate Server with Database and Workstations, Root Server, and Root Server with Workstations. See “Inventory Components on Servers” on page 322 for a quick reference table of server components.

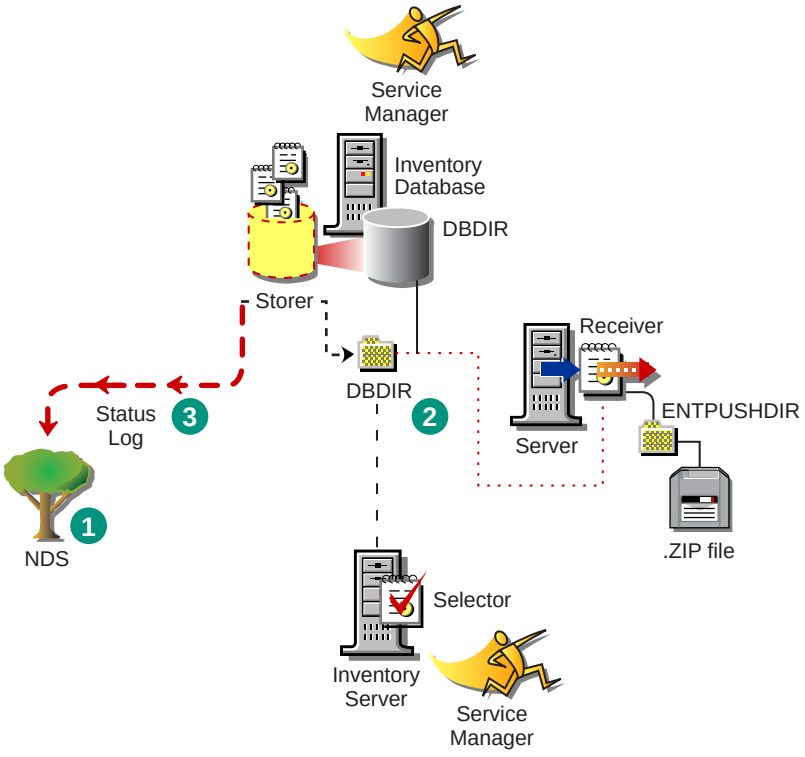
The Storer processes the files in the DBDIR directory. The Storer runs as a Service loaded by the Service Manager.

The processing done by the Storer is as follows:

1. From the server configuration information stored in eDirectory, the Storer looks in the database directory (DBDIR) for the scan files. The 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 scan .ZIP files in DBDIR.
2. 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 .STR file for a workstation.

   The Storer forces a full scan of a workstation if there are problems with the .STR files.
3. The Storer updates the status in the Workstation Status log and updates the Roll-Up log.You can view the status information in the Server Status log. For more information, see “Troubleshooting Workstation Inventory with Status Logs” on page 431.

The following illustration depicts the processing done by the Storer:



Understanding the Service Manager

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

This sections contains the following:

- "List of Services" on page 314
- "Services on NetWare Servers" on page 317
- "Services on Windows NT/2000 Servers" on page 318

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
Receiver Service	Loads the Receiver
Sender Service	Loads the Sender
Storer Service	Loads the Storer

Service Name	Description
Inventory Upgrade Service	Runs the upgrade utilities for inventory
NDSLookupForDB Service	Loads the service for NDSLookupForDB tool
DBDelete Service	Loads the service for DBDelete tool
DBBackup Service	Loads the service for DBBackup tool

The AlterDBSpace tool loads the AlterDBSpace Service and the ZEN2 Delete tool uses the zen2remove Service.

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

A sample role-based property file for 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.UpgradeServiceArgument
s =

[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.SenderServiceI
nit

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 server, the server properties files include:

Server Type	Server Property File
Root Server	ROOT_DB.PROPERTIES

Server Type	Server Property File
Root Server with Workstations	ROOT_DB_WKS.PROPERTIES
Intermediate Server	INT.PROPERTIES
Intermediate Server with Workstations	INT_WKS.PROPERTIES
Intermediate Server with Database	INT_DB.PROPERTIES
Intermediate Server with Database and Workstations	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 \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
```

Services on NetWare Servers

On a NetWare® server, the installation program modifies the ZFDSTART.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 server startup time.

On a NetWare server, the installation program adds the STARTINV.NCF entry in the ZFDSTART.NCF file.

The contents of the ZFDSTART.NCF file are as follows:

```
;ZENWORKS for Desktop 3 Settings

STARTINV.NCF
```

The contents of the STARTINV.NCF file are as follows:

```
InvEnv

Naming

java -DConfigFile=sys:\public\zenworks\wminv\properties\Config.properties -
nsac -jszenWSInv -neh -mx128m -classpath $tmppath;$classpath
com.novell.zenworks.desktop.inventory.servercommon.ZENWorksInventoryService
Manager
```

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

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

This will display the following message:

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

To start a service:

1. At the server prompt, enter **StartSer *service_name***

   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:

1. At the server prompt, enter **StopSer *service_name***

   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 **StopSer *Storer***

   To stop all services, enter **StopSer ***

To list a service:

1. Edit the LISTSER.NCF file located in the SYS:\SYSTEM directory.
2. Modify the *service_name* in the following line:

   ```
   java -neh - classpath $tmppath;CLASSPATH
   com.novell.zenworks.desktop.inventory.servercommon.ZENWorksInventoryList
   Service "service_name"
   ```

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

For example, to list Selector Server, the *service_name* specified in the file should be "Selector Service".

To refer to all services, use the asterisk (*) wildcard character within double quotes "*". This wildcard character can be used with Stop and List services parameters.

Services on Windows NT/2000 Servers

On Windows NT/2000 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 PUBLIC\ZENWORKS\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 PUBLIC\ZENWORKS\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 PUBLIC\ZENWORKS\WMINV\BIN directory.
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.

Inventory Upgrade Service

The Inventory Upgrade Service runs the upgrade utilities for inventory. For example, this service makes the necessary updates in the database tables, and creates database tables. This service triggers full scan on workstations that have more than one MAC Addresses, IP/IPX™ Address, or DNS Name.

This service (Upgrade Service) is auto loaded on NetWare and Windows NT/2000 servers.

Understanding the Effects of Workstation Inventory Installation

On the servers, the Workstation Inventory installation program does the following:

- On NetWare servers:
  - Copies the inventory related files to the PUBLIC\ZENWORKS directory.
  - Copies the Workstation Inventory snap-in component to the PUBLIC\MGMT\CONSOLEONE*ConsoleOne_version*\BIN directory.
  - Assigns Create rights to the SCANDIR directory on NetWare servers.
  - The installation program assigns the SCANDIR as a trustee of the [Root] with Create rights.
  - Creates an Inventory Service object (*servername*_ZenInvservice) in eDirectory for each server on which Inventory is installed. This object is populated with the attributes: zeninvRole (role of the server), zeninvScanFilePath (path to SCANDIR directory), and zeninvHostServer (DN of the server on which Inventory is installed).
  - If the Inventory Service object already exists, the object is destroyed and created again.
  - 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 the tree.

    If there are any higher-level containers preceding the container that has the Inventory Service object, the Inventory Service object is assigned as a trustee of these containers.

    For example, if the Inventory tree is Novell_US, that has an Inventory Service object with the DN as us.california.sanjose.ZENInvService_SanJoseServer, where ZENInvService_SanJoseServer is the object, then ZENInvService_SanJoseServer is made a trustee of Novell_US, us container, california container, and sanjose container.

- Creates the scan directory (SCANDIR) with the subdirectories (ENTPUSHDIR, ENTMERGE, and DBDIR) in the specified volume on the server.
- The PUBLIC\ZENWORKS directory is made a trustee of [Root] with Read and filescan rights.

- On Windows NT/2000 servers:
  - Modifies the registry entries so that at run time, the scanners can be executed on the workstations from the servers. The installation adds the SYS share to the registry entry on Windows NT/2000 servers.
  - Creates the SCANDIR directory with the subdirectories.
  - Creates an Inventory Service object (*servername*_ZenInvservice) in eDirectory for each server on which Inventory is installed. These attributes are populated: zeninvRole (Role of the server), zeninvScanFilePath (Path to ScanDir), and zeninvHostServer (DN of the server on which Inventory is installed).
  - If an Inventory Service object already exists, the object is destroyed and created again.
  - The installation program assigns the Inventory Service object as trustee to the tree.

    If there are any higher-level containers preceeding the container that has the Inventory Service object, the Inventory Service object is assigned as a trustee of these containers.

    For example, if the Inventory tree is Novell_US, that has an Inventory Service object with the DN as us.california.sanjose.ZENInvService_SanJoseServer, where ZENInvService_SanJoseServer is the object, then ZENInvService_SanJoseServer is made a trustee of Novell_US, us container, california container, and sanjose container.
  - Creates the SCANDIR directory (Share directory on the server\ZENWORKS\SCANDIR directory) with the subdirectories (ENTPUSHDIR, ENTMERGE, and DBDIR).
  - The installation program assigns rights to the SCANDIR directory. The installation program modifies the registry entries so that the scanners can create the scan data files (.STR) in the SCANDIR directory.
  - On the server, the Inventory Service Manager is created as a service.
- On the database servers:
  - Installs the Sybase* ASA 7.0.0.505 database on the server you specify.
  - On a NetWare server, if Sybase 7 is already installed, only the database files are copied to the server.
  - If the Sybase database does not exist on the NetWare server, then Sybase database and database files are copied. Also, the installation program creates the MGMTDBS.NCF file in the SYS:\SYSTEM directory.
  - On Windows NT/2000 servers, the Sybase database and the database files are copied. Also, the database server service (Adaptive Service Anywhere - ZENworks for Desktops 3) is created.
  - On the servers, the installation program initializes the database. At server startup time, the database is loaded.
  - If the MGMTDBS.NCF file already exists, then MGMTDB.DB and NAL.DB entries are added to this file, if these entries do not exist. Also, any other entries that ZfD uses are added if these entries do not exist.

- Creates a Database object (ZfDInventoryDatabase) for Sybase and configures the properties of the object.

- On an existing ZENworks 2 installation:
  - On a NetWare server, detects whether a ZENworks 2 installation exists. If so, the existing directory under \SYS:\PUBLIC is renamed as ZENWORKS.ZEN2BKUP*currentdate*. Also, ZfD3 installation takes care of loading the Inventory database (Sybase) for ZENworks 2 and ZfD3.
  - Detects if an existing Sybase database with ZENworks 2 or ZENworks for Servers (ZfS) exists. If so, the installation program checks the database version and file size. If the existing Sybase version is 7.0.0.313, the installation program copies the new database and does not copy Sybase files.
  - If ZENworks 2 with Support Pack 1 exists on the server, the MGMTDB.DB and NAL.DB entries are added to MGMTDBS.NCF file. The ZENworks 2 entries in the AUTOEXEC.NCF entries will be deleted.
  - If a ZENworks 2 setup exists on the server, the ZENINVDB entry is added to the MGMTDBS.NCF file. The ZENworks 2 entries in the AUTOEXEC.NCF entries will be deleted.
  - ZENworks 2 inventory entries are removed from the AUTOEXEC.NCF file.
- On an existing ZfD 3 installation:
  - Detects if ZfD 3 files exist. If so, the files are overwritten. The existing Inventory Service object and the database object are overwritten. You need to configure the policies and properties for the new Inventory Service object and the database object.
- On the workstations:

  When you install the Novell Client™ on the workstation, the following files are copied on the workstation:

Filename	Directory
PUBLIC\ZENWORKS\WM95INV3.DLL	WINDOWS\SYSTEM on Windows 95/98
PUBLIC\ZENWORKS\WMINV3.DLL	WINNT\SYSTEM32 on Windows NT/2000
PUBLIC\ZENWORKS\INVSTAT.DLL	WINDOWS\SYSTEM on Windows 95/98
	WINNT\SYSTEM32 on Windows NT/2000
PUBLIC\ZENWORKS\LOC32VC.DLL	WINDOWS\SYSTEM on Windows 95/98
	WINNT\SYSTEM32 on Windows NT/2000
PUBLIC\ZENWORKS\NLS\ENGLISH\INVMSG.DLL	WINDOWS\SYSTEM\NLS\ENGLISH on Windows 95/98
	WINNT\SYSTEM32\NLS\ENGLISH on Windows NT/2000
PUBLIC\ZENWORKS\ZENPOL32.DLL	WINDOWS\SYSTEM on Win 95/98
	WINNT\SYSTEM32 on Win NT/2000

Inventory Components on Servers

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

Server Component	Root Server	Leaf Server with Database	Leaf Server	Intermediate Server	Intermediate Server with Database and Workstations	Intermediate Server with Database	Standalone Server	Root Server with Workstations
Service Manager	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Selector	--	Yes	Yes	--	Yes	--	Yes	Yes
Storer	Yes	Yes	--	--	Yes	Yes	Yes	Yes
Sender	--	Yes	Yes	Yes	Yes	Yes	--	--
Receiver	Yes	--	--	Yes	Yes	Yes	--	Yes
Database	Yes	Yes	--	--	Yes	Yes	Yes	Yes

Understanding ZfD Inventory Attributes

The following table lists the Workstation Inventory attributes that ZENworks for Desktops 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)
- 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.

For more information about the ZfD Inventory database, see "Understanding the ZENworks for Desktops Inventory Database Schema" on page 344.

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 workstation registered in eDirectory
General-Asset-Description	Asset_Description	Zenworks.SystemInfo.Description	String	254	Description of the system asset information
General-Asset-Caption	Asset_Caption	Zenworks.SystemInfo.Caption	String	64	Identifying information of the computer

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-Asset-Asset Tag	Asset_AssetTag	Zenworks.SystemInfo.AssetTag	String	256	Asset tag number that the ROM-based setup program creates. This is unique to every workstation.
General-Asset-Model Number	Asset_Model Number	Zenworks.SystemInfo.Model	String	64	Model number value for the computer, assigned during manufacture
General-Asset-Serial Number	Asset_SerialNumber	Zenworks.SystemInfo.Serial Number	String	64	Model serial number value for the computer, assigned during manufacture
General-User Information-Current Login User	UserInformation_Currentloginuser	ManageWise."User".CurrentLoginName	String	254	User logged in to the Primary eDirectory tree when the workstation was scanned
General-User Information.-Last Login User	UserInformation_Lastloginuser	ManageWise."User".LastLogin Name	String	254	User logged in last to the Primary eDirectory tree when the workstation was scanned
SOFTWARE - Applications-Name	Applications_Name	CIM.Product.Name	String	256	Name of the software application
SOFTWARE - Applications-Vendor	Applications_Vendor	CIM.Product.Vendor	String	256	Name of the software application manufacturer
SOFTWARE - Applications-Version	Applications_Version	CIM.Product.Version	String	64	Version of the software application
SOFTWARE-Operating Systems-Name	OperatingSystems_Name	ZENworks.ZENOperatingSystem.OSType	Unsigned Small Integer (enum)		Operating system name. For example, Windows NT/ Windows 2000. See “Enumeration Values for SOFTWARE-Operating Systems-Name” on page 343.
SOFTWARE-Operating Systems-Version	OperatingSystems_Version	ZENworks.ZENOperatingSystem.Version	String	254	Version of the 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
SOFTWARE-Operating Systems-Description	OperatingSystems_Description	ZENworks.ZENOperatingSystem.Description	String	254	More description about the operating system
SOFTWARE-Operating Systems-Install Date	OperatingSystems_InstallDate	ZENworks.ZENOperatingSystem.InstallDate	String	25	Install date of the operating system
SOFTWARE-Operating Systems-Size Stored in Paging Files	OperatingSystems_SizeStoredInPagingFiles(Kb)	ZENworks.ZENOperatingSystem.SizeStoredInPagingFiles	Unsigned Bit Integer		Total size allocated for paging by the operating system
SOFTWARE-Operating Systems-Code Page	OperatingSystems_CodePage	ZENworks.ZENOperatingSystem.CodePage	String	254	Current language code page being used
SOFTWARE-Scanner Information-Version	ScannerInformation_Version	ZENworks.InventoryScanner Version	String	64	Version of the scanner running on the workstation
SOFTWARE-Scanner Information-Last Scan Date	ScannerInformation_LastScanDate	ZENworks.InventoryScanner. LastScanDate	Unsigned Bit 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
SOFTWARE-Scanner Information-Inventory Server	ScannerInformation_InventoryServer	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
SOFTWARE-NetWare Client-Version	NetwareClient_Version	ZENworks.NetWareClient.Version	String	64	Version of the NetWare client software installed on the workstation
SOFTWARE-Drivers-Network Adapter Driver-Description	NetworkAdapterDriver_Description	ZENworks.NetworkAdapterDriver.Description	String	254	Description of the network adapter driver installed on the workstation. For example, IBM 10/100 Ethernet adapter, EN-2420Px Ethernet 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
SOFTWARE-Drivers-Network Adapter Driver-Name	NetworkAdapterDriver_Name	ZENworks.NetworkAdapterDriver.Name	String	256	Name of the network adapter driver software installed, that corresponds to the adapter. For example, ne2000.sys, pppmac.vxd and others
SOFTWARE-Drivers-Network Adapter Driver-Version	NetworkAdapterDriver_Version	ZENworks.NetworkAdapterDriver.Version	String	64	Network adapter driver version
SOFTWARE-Drivers-Mouse Driver-Name	MouseDriver_Name	ZENworks.PointingDeviceDeviceDriver.Name	String	256	Name of the mouse driver installed on the workstation
SOFTWARE-Drivers-Mouse Driver-Version	MouseDriver_Version	ZENworks.PointingDeviceDeviceDriver.Version	String	64	Mouse driver version
HARDWARE-Mouse-Name	Mouse_Name	CIM.PointingDevice.Name	String	256	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 341.
HARDWARE-Mouse-Number of buttons	Mouse_NumberofButtons	CIM.PointingDevice.NumberOfButtons	Unsigned Tiny Integer		The number of buttons used by the pointing device

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Mouse-IRQ Number	Mouse_IRQNumber	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 Desktops Inventory Database Schema" on page 344.
HARDWARE-Keyboard.Number of function keys	Keyboard_NumberofFunctionKeys	ZENworks.ZENKeyboard.NumberOfFunctionKeys	UnsignedSmall Integer		Number of function keys on keyboard
HARDWARE-Keyboard.Layout	Keyboard_Layout	ZENworks.ZENKeyboard.layout	String	254	Layout information. For example, US English.
HARDWARE-Keyboard. SubType	Keyboard_Subtype	ZENworks.ZENKeyboard.SubType	Unsigned Integer		A number indicating the subtype of the keyboard
HARDWARE-Keyboard.Delay	Keyboard_Delay(mSecs)	ZENworks.ZENKeyboard.Delay	Unsigned Integer		Delay before the repeat of a key
HARDWARE-Keyboard. Typematic rate	Keyboard_TypematicRate (mSecs)	ZENworks.ZENKeyboard.Typematic Rate	Unsigned Integer		Rate of processing the keys
HARDWARE-Keyboard. Description	Keyboard_ Description	ZENworks. ZENKeyboard.Description	String	254	Keyboard description indicating the type of keyboard. For example, IBM* enhanced (101/102 key) keyboard.
HARDWARE-Video BIOS. Manufacturer	VideoBIOS_ Manufacturer	CIM.Video BIOSElement.Manufacturer	String	254	Manufacturer of the video BIOS driver installed on the system
HARDWARE-Video BIOS.Version	Video BIOS_Version	CIM.Video BIOSElement.Version	String	254	Version of the Video BIOS driver
HARDWARE-Video BIOS.Install Date	Video BIOS_InstallDate	CIM.Video BIOSElement.InstallDate	String	25	Video BIOS release date

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Video BIOS.IS Shadowed	Video BIOS_IsShadowed	CIM.Video BIOSElement.ISShadowed	BIT (Used for Boolean conditionshere)		Is a Boolean condition indicating the video BIOS supports Shadow memory. 0 represents false and 1 is true.
HARDWARE-Display Adapter.Number Of color panes	DisplayAdapter_ NumberofColor Planes	ZENworks. VideoAdpater.NumberOf ColorPanes	Unsigned Integer		Number of color planes supported by the video system
HARDWARE-Display Adapter.Current Vertical Resolution	DisplayAdapter_ CurrentVertica lResolution	ZENworks. VideoAdpater.Current Vertical Resolution	Unsigned Integer		Vertical resolution of the display
HARDWARE-Display Adapter.Current Horizontal Resolution	DisplayAdapter_ CurrentHorizontalResoluti on	ZENworks. VideoAdpater.Current Horizontal Resolution	Unsigned Integer		Horizontal resolution of the display
HARDWARE-Display Adapter. Description	DisplayAdapter_ Description	ZENworks. VideoAdpater.Description	String	254	Video adapter description
HARDWARE-Display Adapter.Min Refresh Rate	DisplayAdapter_ Min_RefreshRate	ZENworks. VideoAdpater.MinRefresh Rate	Unsigned Integer		Minimum refresh rate of the monitor for redrawing the display, measured in Hertz
HARDWARE-Display Adapter.Max Refresh Rate	DisplayAdapter_ Max_RefreshRate	ZENworks. VideoAdpater.MaxRefresh Rate	Unsigned Integer		Maximum refresh rate of the monitor for redrawing the display, measured in Hertz
HARDWARE-Display Adapter.Video Architecture	DisplayAdapter_ VideoArchitecture	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 340.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Display Adapter.Video Memory Type	DisplayAdapter_ VideoMemoryType	ZENworks. VideoAdpater.VideoMemoryType	Unsigned Small Integer (Enum)		The type of memory for this adapter. For example, VRAM/ SRAM/DRAM/EDO RAM. See Enumeration Values for HARDWARE-Display Adapter.Video Memory Type.
HARDWARE-Display Adapter.Max memory supported	DisplayAdapter_ MaxMemory Supported(bytes)	ZENworks. VideoAdpater.MaxMemory Supported	Unsigned Integer		Maximum memory that the display adapter supports for VIDEO RAM
HARDWARE-Display Adapter.Current Bits/Pixel	DisplayAdapter_ CurrentBits/Pixel	ZENworks. VideoAdpater.CurrentBits PerPixel	Unsigned Integer		Number of adjacent color bits for each pixel
HARDWARE-Modem.Caption	Modem_Caption	CIM.POTS Modem. Caption	String	64	Modem label. Usually the name of the manufacturer is stored here. For example, 3Com, IBM*.
HARDWARE-Modem. Description	Modem_Description	CIM.POTS Modem. Description	String	254	The complete description of the modem. For example, Standard 2400 bps modem, IBM PCMCIA HPC modem.
HARDWARE-Modem.Name	Modem_Name	CIM.POTS Modem.Name	String	256	The name of the modem dictating its type and usage. For example, Standard Windows Modem means that this is used in standard windows architecture.
HARDWARE-BIOS.BIOS Identification Bytes	BIOS_BIOS IdentificationBytes	ZENworks. BIOS.BIOS IDBytes	String	254	Byte in the BIOS that indicates the computer model
HARDWARE-BIOS.Serial Number	BIOS_ SerialNumber	ZENworks. BIOS.Serial Number	String	64	Serial number of BIOS assigned by the manufacturer

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-BIOS.Primary BIOS	BIOS_PrimaryBios	ZENworks. BIOS.PrimaryBIOS	BIT (Used for Boolean conditions here)		True when set to 1, indicates that this BIOS is the primary BIOS. Used in systems with additional BIOS chips.
HARDWARE-BIOS.Install Date	BIOS_InstallDate	ZENworks. BIOS.Install Date	String	25	The release date of the BIOS given by the manufacturer
HARDWARE-BIOS.Version	BIOS_Version	ZENworks. BIOS.Version	String	254	Version or revision level of the BIOS
HARDWARE-BIOS. Manufacturer	BIOS_ Manufacturer	ZENworks. BIOS. Manufacturer	String	254	The manufacturer name of BIOS
HARDWARE-BIOS.Caption	BIOS_Caption	ZENworks. BIOS.Caption	String	64	The name of the BIOS as given by the BIOS manufacturer
HARDWARE-BIOS.size	BIOS_Size(KB)	ZENworks. BIOS.size	Unsigned Integer		Size of the BIOS in bytes
HARDWARE-Processor-Current Clock Speed	Processor_CurrentClockSpeed(MHz)	CIM. Processor. CurrentClockSpeed	Unsigned Integer		Current clock speed of the processor in MHz
HARDWARE-Processor-Maximum clock speed	Processor_ MaximumClock Speed (MHz)	CIM. Processor. MaxClock Speed	Unsigned Integer		Maximum clock speed of the processor in MHz
HARDWARE-Processor-Role	Processor_Role	CIM. Processor. Role	String	254	Type of processor such as central processor, math coprocessor, and others
HARDWARE-Processor-Processor Family	Processor_ ProcessorFamily	CIM. Processor. Family	Unsigned Small Integer (enum)		Family the processor belongs to. See "Enumeration Values for HARDWARE-Processor-Processor Family" on page 341.
HARDWARE-Processor-Other family description	Processor_Other FamilyDescription	CIM. Processor. OtherFamily Description	String	64	Additional description about the processor family, such as the Pentium* processor with MMX technology

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Processor-Upgrade Method	Processor_ UpgradeMethod	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 342.
HARDWARE-Processor - Processor Stepping	Processor_ ProcessorStepping	CIM. Processor. Stepping	String	254	Single-byte code characteristic provided by microprocessor vendors to identify the processor stepping model
HARDWARE-Processor-Device ID	Processor_ DeviceID	CIM. Processor. DeviceID	String	64	Special hexadecimal string identifying the processor type
HARDWARE-Memory-Physical Memory-Physical Memory Size	PhysicalMemory_Physical Memory Size(Kb)	CIM.Memory.NumberOf Blocks	Unsigned Bit Integer		Total physical memory size of the workstation
HARDWARE-Memory-Virtual Memory-Total Virtual Memory	VirtualMemory_ TotalVirtual Memory(Kb)	ZENworks. VirtualMemory.TotalVirtual MemorySize			Indicates the total number of bytes in the virtual address space
HARDWARE-Memory-Cache Memory-Speed	CacheMemory_ Speed(nsec)	CIM.PhysicalMemory. Speed	Unsigned Bit 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 Desktops Inventory Database Schema" on page 344.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Capacity	CacheMemory_ Capacity(Kb)	CIM.PhysicalMemory. Capacity	Unsigned Bit 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 Desktops Inventory Database Schema" on page 344.
HARDWARE-Memory-Cache Memory-Level	CacheMemory_ 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. Enumeration values:1 = "Other" 2 = "Unknown" 3 = "Primary" 4 = "Secondary" 5 = "Tertiary" 6 = "Not Applicable"
HARDWARE-Memory-Cache Memory-Write Policy	CacheMemory_ WritePolicy	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. Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Write Back" 4 = "Write Through" 5 = "Varies with Address" 6 = "Determination Per I/O"

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Error methodology	CacheMemory_ ErrorMethodology	CIM.CacheMemory.Error Methodology	String	254	Error correction scheme supported by this cache component, for example, Parity/ Single Bit ECC/ MultiBit ECC
HARDWARE-Memory-Cache Memory-Cache type	CacheMemory_ CacheType	CIM.Cache Type	Unsigned Small Integer (enum)		Defines the system cache type. For example, Instruction, Data, Unified. Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Instruction" 4 = "Data" 5 = "Unified"
HARDWARE-Memory-Cache Memory-Line Size	CacheMemory_ LineSize(Bytes)	CIM.Cache Memory .LineSize	Unsigned Integer		Size in bytes of a single cache bucket or line
HARDWARE-Memory-Cache Memory-Replacement Policy	CacheMemory_ 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 342.
HARDWARE-Memory-Cache Memory-Read Policy	CacheMemory_ ReadPolicy	CIM.Cache Memory. ReadPolicy	Unsigned Small Integer (enum)		Indicates whether the data cache is for read operation.Enumeration values are as follows: 1 = "Other" 2 = "Unknown" 3 = "Read" 4 = "Read-ahead" 5 = "Read and Read-ahead" 6 = "Determination Per I/O"

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Associativity	CacheMemory_ Associativity	CIM.Cache Memory. Associativity	Unsigned Integer (enum)		Defines the system cache associativity (direct-mapped, 2-way, 4-way)Enumeration values are as follows: 1 = "Other" 2 = "Unknown" 3 = "Direct Mapped" 4 = "2-way Set-Associative" 5 = "4-way Set-Associative" 6 = "Fully Associative"
HARDWARE-Disk-Floppy-Manufacturer	Floppy_ Manufacturer	ZENworks. Physical Diskette. Manufacturer	String	256	Vendor name
HARDWARE-Disk-Floppy-. Description	Floppy_ Description	ZENworks. Physical Diskette. Description	String	254	Floppy diskette description
HARDWARE-Disk-Floppy-Physical Cylinders	Floppy_Physical Cylinders	ZENworks. Physical Diskette. Physical Cylinders	Unsigned Integer		Total number of cylinders or tracks on the floppy
HARDWARE-Disk- Floppy - Physical Heads	Floppy_Physical Heads	ZENworks. Physical Diskette. Physical Heads	Unsigned Small Integer		Number of heads
HARDWARE-Disk- Floppy - Capacity	Floppy_Capacity (Kb)	ZENworks. Physical Diskette. Capacity	Unsigned Bit Integer		Total size
HARDWARE-Disk- Floppy - Sectors/Track	Floppy_Sectors/Track	ZENworks. Physical Diskette. SectorsPer Track	Unsigned Integer		Number of sectors per track
HARDWARE-Disk-Floppy Drive- DeviceID	FloppyDrive_ DeviceID	CIM.Diskette Drive	String	64	The drive name representing the floppy drive
HARDWARE-Disk-Hard Disk-Manufacturer	HardDisk_ Manufacturer	ZENworks. PhysicalDisk.Manufacturer	String	256	Vendor name

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Disk- Hard Disk - .Description	HardDisk_ Description	ZENworks. PhysicalDisk.Description	String	254	Hard disk vendor description
HARDWARE-Disk- Hard Disk - Physical Cylinders	HardDisk_ PhysicalCylinders	ZENworks. PhysicalDisk.Physical Cylinders	Unsigned Integer		Total number of cylinders
HARDWARE-Disk- Hard Disk - Physical Heads	HardDisk_PhysicalHeads	ZENworks. PhysicalDisk.Physical Heads	Unsigned Small Integer		Number of heads
HARDWARE-Disk- Hard Disk - Sectors/Track	HardDisk_Sectors/Track	ZENworks. PhysicalDisk.SectorsPer Track	Unsigned Integer		Number of sectors per track
HARDWARE-Disk- Hard Disk - Capacity	HardDisk_ Capacity(Kb)	ZENworks. PhysicalDisk.Capacity	Unsigned Bit Integer		Total size of the hard disk
HARDWARE-Disk-Hard Disk Drive-Device ID	HardDiskDrive_ DeviceID	ZENworks. LogicalDisk Drive. DeviceID	String	64	The logical drive letter partitioned on the hard disk. For example, C:, D:, and others.
HARDWARE-Disk-Hard Disk Drive-Size	HardDiskDrive_ Size(MB)	ZENworks. LogicalDisk Drive. TotalSize	Unsigned Integer		Total size on the logical drive
HARDWARE-Disk Hard Disk Drive-Available Space	HardDiskDrive_ FreeSize(MB)	ZENworks. LogicalDisk Drive. FreeSize	Unsigned Integer		Available space on the logical drive
HARDWARE-Disk-Hard Disk Drive-Volume Serial Number	HardDiskDrive_ VolumeSerial Number	ZENworks. LogicalDisk Drive.VolumeSerialNumber	String	254	Volume serial number of the logical drive
HARDWARE-Disk-Hard Disk Drive-Caption	HardDiskDrive_ Caption	ZENworks. LogicalDisk Drive.Caption	String	64	Volume label assigned to the logical drive
HARDWARE-Disk-CDROM-Name	CDROM_Name	ZENworks. Physical CDROM. Manufacturer	String	256	The manufacturer of the CD-ROM drive
HARDWARE-Disk-CDROM-Caption	CDROM_Caption	ZENworks. Physical CDROM. Caption	String	64	CD-ROM label

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Disk-CDROM-Description	CDROM_ Description	ZENworks. Physical CDROM. Description	String	256	Description of the CD-ROM drive, as given by the manufacturer. For example, ATAPI CDROM, CREATIVE CD1620E SL970520.
HARDWARE-Disk-CDROM Drive-DeviceID	CDROMDrive_ DeviceID	ZENworks. Logical CDROM. DeviceID	String	64	Drive letter allocated for the CD-ROM on the workstation
HARDWARE-Disk-Backup Disk-Name	BackUpDisk_ Name	ZENworks. SCSIDrive. Name	String	256	Name of the backup device vendor. For example, IOMEGA.
HARDWARE-Disk-Backup Disk-Description	BackUpDisk_ Description	ZENworks. SCSIDrive. Description	String	254	The description of the backup device or type. For example, IOMEGA ZIP 100 indicates that it is a zip drive.
HARDWARE-Disk-Backup Disk Drive-DeviceID	BackUpDiskDrive_DeviceID	ZENworks. Logical SCSIDrive-DeviceID	String	64	The drive mapped to the backup disk on the workstation
HARDWARE-Ports-Serial Port-Name	SerialPort_Name	ZENworks. SerialPort. Name	String	256	The name of the serial port. For example, COM1, COM2 and others.
HARDWARE-Ports-Serial Port-Address	SerialPort_ Address	ZENworks. SerialPort. Address	Unsigned Bit 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
HARDWARE-Ports-Serial Port-IRQ	SerialPort_IRQ	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 "Understanding the ZENworks for Desktops Inventory Database Schema" on page 344.
HARDWARE-Ports-Parallel Port-Name	ParallelPort_Name	ZENworks. ParallelPort. Name	String	256	The name of the parallel port. For example, LPT1 and others.
HARDWARE-Ports-Parallel Port-Address	ParallelPort_ Address	ZENworks. ParallelPort. Address	Unsigned Bit Integer		The name of the parallel port. For example, LPT1 and others
HARDWARE-Ports-Parallel Port-DMA Support	ParallelPort_DMASupport	ZENworks. ParallelPort. DMASupport	BIT (used for Boolean conditions 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
HARDWARE-Ports-Parallel Port-IRQ	ParallelPort_IRQ	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 parallel Port class in the database. For more information on how they are associated, see "Understanding the ZENworks for Desktops Inventory Database Schema" on page 344.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Bus-Version	Bus_Version	ZENworks. Bus.Bus Version	String	254	Version of the Bus supported by the motherboard
HARDWARE-Bus-Protocol Supported	Bus_Protocol Supported	ZENworks. Bus. Protocol Supported	Unsigned Small Integer (Enum)		Bus type indicates PCI, ISA, and others. See "Enumeration Values for HARDWARE-Bus-Protocol Supported" on page 343.
HARDWARE-Network Adapter-Name	NetworkAdapter_ Name	CIM.EthernetAdapter.Name	String	256	Network adapters installed on the system
HARDWARE-Network Adapter-Max. Speed	NetworkAdapter_Max_Speed(bits/sec_)	CIM.EthernetAdpater. MaxSpeed	Unsigned Bit Integer		Rate at which the adapter can transfer data
HARDWARE-Network Adapter-Permanent Address	NetworkAdapter_ Permanent Address	CIM.EthernetAdapter. Permanent Address	String	64	Machine address stored permanently in the adapter (MAC address)
HARDWARE-MultiMedia Card-Description	MultimediaCard_ Description	ZENworks. SoundCard. Description	String	254	Description of the multimedia component for the workstation
HARDWARE-MultiMedia Card-Name	MultimediaCard_ Name	ZENworks. SoundCard. Name	String	256	Name of the sound card installed on the system
HARDWARE-MultiMedia Card-Manufacturer	MultimediaCard_ Manufacturer	ZENworks. SoundCard. Manufacturer	String	256	Vendor name
HARDWARE-Cards-Description	Cards_Description	CIM.Card. Description	String	254	Name of the cards installed on the workstation
HARDWARE-Battery-Name	Battery_Name	CIM.Battery. Name	String	254	Name of the battery installed on the system
HARDWARE-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 341.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Battery-Design Capacity	Battery_Design Capacity(mWatt-hours)	CIM.Battery. Design Capacity	Unsigned Integer		The design capacity of the battery in mWatt-hours
HARDWARE-Battery-Design Voltage	Battery_Design Voltage(MilliVolts)	CIM.Battery. DesignVoltage	Unsigned Bit Integer		The design voltage of the battery in mVolts
HARDWARE-Battery-Smart Battery Version	Battery_ SmartBattery Version	CIM.Battery. SmartBatteryVersion	String	64	The Smart Battery Data Specification version number supported by this battery
HARDWARE-Battery-Manufacturer	Battery_ Manufacturer	CIM.PhysicalComponent. Manufacturer	String	254	Vendor name of the battery
HARDWARE-Battery-Install Date	Battery_Install Date	CIM.PhysicalComponent. InstallDate	String	25	Date of manufacturing the battery
HARDWARE-Battery-Serial Number	Battery_Serial Number	CIM.PhysicalComponent. SerialNumber	String	64	Battery serial number
HARDWARE-Power Supply-Description	PowerSupply_ Description	CIM.Power Supply. Description	String	254	Name and description of the power supply on the system
HARDWARE-Power Supply-Total Output Power	PowerSupply_TotalOutput Power (MilliWatts)	CIM.Power Supply.Total OutputPower	Unsigned Integer		Total output power of the power supply
NETWORK-IP Address - Address	IPAddress_ Address	CIM.IP Protocol Endpoint. Address	String	254	IP address of the workstation
NETWORK-IP Address - Subnet Mask	IPAddress_ SubnetMask	CIM.IP Protocol Endpoint. SubnetMask	String	254	The subnet mask of the workstation
NETWORK -DNS - LABEL	DNS_LABEL	ManageWise.DNSName. Label	String	254	DNS name of the workstation
NETWORK -IPX Address - Address	IPXAddress_ Address	CIM.IPX Protocol Endpoint. Address	String	254	IPX address of the workstation
NETWORK -Mac Address - MACAddress	MacAddress_ MACAddress	CIM.LAN Endpoint. MACAddress	String	12	MAC address of the workstation

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SYSTEM-MotherBoard-Hosting Board	MotherBoard_ HostingBoard	CIM.Card. HostingBoard	BIT(used for Boolean conditions)		When True or 1, the card represents a motherboard. For all other cards, this attributes will be False or 0
SYSTEM-MotherBoard-Version	MotherBoard_ Version	CIM.Card. Version	String	64	Motherboard version
SYSTEM-MotherBoard-Description	MotherBoard_ Description	CIM.Card. Description	String	254	The number of expansion slots on the motherboard
SYSTEM-IRQ-IRQ Number	IRQ_IRQNumber	CIM.IRQ.IRQNumber	Unsigned Integer		The system interrupt number
SYSTEM-IRQ-Availability	IRQ_ IRQAvailability	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"
SYSTEM-IRQ-IRQ Trigger Type	IRQ_IRQTrigger 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"
SYSTEM-IRQ-IRQ Shareable	IRQ_IRQ Shareable	CIM.IRQ. Shareable	Unsigned Small Integer		Boolean indicating whether the IRQ can be shared

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SYSTEM-SLOT-Max. Data Width	Slot_Max_ DataWidth	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
SYSTEM-SLOT-Thermal Rating	Slot_Thermal Rating(MilliWatts)	CIM.Slot. Thermal Rating	Unsigned Integer		Maximum thermal dissipation of the slot in milliwatts
SYSTEM-DMA-DMA Channel Number	DMA_DMA ChannelNumber	CIM.DMA. DMAChannel	Unsigned Integer		The DMA channel number
SYSTEM-DMA-Description	DMA_Description	CIM.DMA. Description	String	254	The name of the device using the DMA channel
SYSTEM-DMA-Availability	DMA_IRQ 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"
SYSTEM-DMA-DMA Burst Mode	DMA_DMA BurstMode	CIM.DMA. BurstMode	BIT (used for Boolean condition here)		Indication that the DMA channel supports the burst mode

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"	

See “HARDWARE-Display Adapter.Video Architecture” on page 327.

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"	

See “HARDWARE-Display Adapter.Video Memory Type” on page 328.

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"

See “HARDWARE-Mouse-Name” on page 325.

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"

See “HARDWARE-Battery-Chemistry” on page 337.

Enumeration Values for HARDWARE-Processor-Processor Family

The enumeration values are:

1 = "Other"	36 = "Power PC 604"	98 = "68010"
2 = "Unknown"	38 = "Power PC X704"	99 = "68000"
3 = "8086"	39 = "Power PC 750"	100 = "68020"
4 = "80286"	48 = "Alpha Family"	101 = "68030"
5 = "80386"	50 = "Alpha 21066"	112 = "Hobbit Family"
6 = "80486	51 = "Alpha 21164"	128 = "Weitek"

7 = "8087"	52 = "Alpha 21164PC"	144 = "PA-RISC Family"
8 = "80287"	53 = "Alpha 21164a"	145 = "PA-RISC 8500"
9 = "80387"	55 = "Alpha 21364"	146 = "PA-RISC
10 = "80487"	64 = "MIPS Family"	147 = "PA-RISC 7300LC"
11 = "Pentium Family"	65 = "MIPS R4000"	147 = "PA-RISC 7300LC"
12 = "Pentium Pro"	66 = "MIPS R4200"	148 = "PA-RISC 7200"
13 = "Pentium II"	67 = "MIPS R4400"	149 = "PA-RISC 7100LC"
14 = "Pentium MMX"	68 = "MIPS R4600"	150 = "PA-RISC 7100"
15 = "Celeron"	69 = "MIPS R10000"	160 = "V30 Family"
16 = "Pentium II Xeon"	80 = "SPARC Family"	176 ="Pentium III Xeon"
18 = "M1 Family"	81 = "SuperSPARC"	180 = "AS400 Family"
26 = "K6 Family"	82 = "microSPARC II"	200 = "IBM390 Family"
27 = "K6-2"	83 = "microSPARC IIep"	201 = "G4"
28 = "K6-3"	84 = "UltraSPARC"	202 = " 65"
29 = "K7"	85 = "UltraSPARC II"	250 = "i860"
30 = "AMD29000 Family"	86 = "UltraSPARC IIi"	251 = "i960"
32 = "Power PC Family"	87 = "UltraSPARC III"	280 = "ARM"
33 = "Power PC 601"	88 = "UltraSPARC IIIi"	281 = "StrongARM"
34 = "Power PC 603"	96 = "68040"	300 = "6x86"
35 = "Power PC 603+"	97 = "68xxx Family"	301 = "MediaGX"
302 = "WinChip"	350 = "DSP"	500 = "Video Processor"

See "HARDWARE-Processor-Processor Family" on page 329.

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"

See "HARDWARE-Processor-Upgrade Method" on page 330.

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

The enumeration values are:

1 = "Other"	5 = "Last In First Out (LIFO)"

2 = "Unknown"

3 = "Least Recently Used (LRU)"

4 = "First In First Out (FIFO)"

6 = "Least Frequently Used (LFU)"

7 = "Most Frequently Used (MFU)"

8 = "Data Dependent Multiple Algorithm"

See "HARDWARE-Memory-Cache Memory-Replacement Policy" on page 332.

Enumeration Values for SOFTWARE-Operating Systems-Name

The enumeration values are:

1 = "Other"

2 = "MACOS"

3 = "ATTUNIX"

4 = "DGUX"

5 = "DECNT"

6 = "Digital Unix"

7 = "OpenVMS"

8 = "HPUX"

9 = "AIX"

10 = "MVS"

11 = "OS400"

12 = "OS/2"

13 = "JavaVM"

14 = "MSDOS"

15 = "WIN3x"

16 = "WIN95"

17 = "WIN98"

18 = " WINNT"

21 = "NetWare"

22 = "OS"

23 = "DC/OS"

24 = "Reliant UNIX"

25 = "SCO UnixWare"

26 = "SCO OpenServer"

27 = "Sequent"

28 = "IRIX"

29 = "Solaris"

30 = "SunOS"

31 = "U6000"

32 = "ASERIES"

33 = "TandemNSK"

34 = "TandemNT"

35 = "BS2000"

36 = "Linux"

37 = "Lynx"

38 = "XENIX"

39 = "VM/ESA"

40 = "Interactive UNIX"

41 = "BSDUNIX"

43 = "NetBSD"

44 = "GNU Hurd"

45 = "OSg"

46 = "MACH Kernel"

47 = "Inferno"

48 = "QNX"

49 = "EPOC"

50 = "IxWorks"

51 = "VxWorks"

52 = "MiNT"

53 = "BeOS"

54 = "HP MPE"

55 = "NextStep"

56 = "PalmPilot"

57 = "Rhapsody"

58 = "Windows 2000"

59 = "Dedicated"

60 = "OS/390"

61 = "VSE"

62 = "TPF"

63 = "Windows (R) Me"

See "SOFTWARE-Operating Systems-Name" on page 323.

Enumeration Values for HARDWARE-Bus-Protocol Supported

The enumeration values are:

1 = "Other"

7 = "Flexible Diskette"

8 = "1496"

9 = "SCSI Parallel Interface"

21 = "Power"

22 = "HIPPI"

23 = "MultiBus"

24 = "VME"

36 = "ESDI"

37 = "IDE"

38 = "CMD"

39 = "ST506"

10 = "SCSI Fibre Channel Protocol""	25 = "IPI"	40 = "DSSI"
11 = "SCSI Serial Bus Protocol"	26 = "IEEE-488"	41 = "QIC2"
12 = "SCSI Serial Bus Protocol - 2 (1394)	27 = "RS232"	42 = "Enhanced ATA/IDE"
13 = "SCSI Serial Storage Architecture"	28 = "IEEE 802.3 10BASE5"	43 = "AGP"
14 = "VESA"	29 = "IEEE 802.3 10BASE2"	44 = "TWIRP (two-way infrared"
15 = "PCMCIA"	30= "IEEE 802.3. 1BASE5"	45 = "FIR (fast infrared)"
16 = "Universal Serial Bus"	31 = "IEEE 802.3 10BROAD36"	46 = "SIR (serial infrared)"
17 = "Parallel Protocol"	32 = "IEEE 802.3 100BASEVG"	47 = "IrBus"
18 = "ESCON"	33 = "IEEE 802.5 Token-Rin"	
19 = "Diagnostic"	34 = "ANSI X3T9.5 FDDI"	
20 = "12C"	35 = "MCA"	

See "HARDWARE-Bus-Protocol Supported" on page 337.

Understanding the ZENworks for Desktops Inventory Database Schema

This section describes the design of the ZfD Inventory database schema implemented using the Common Information Model (CIM) of Distributed Management Task Force (DMTF). To understand this document effectively, you should be familiar with terminology such as CIM and Desktop Management Interface (DMI). You should also have a solid understanding of relational database based managed systems (RDBMS) and database concepts.

The following sections provide in-depth information:

Overview

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

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

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

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

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

CIM Schema

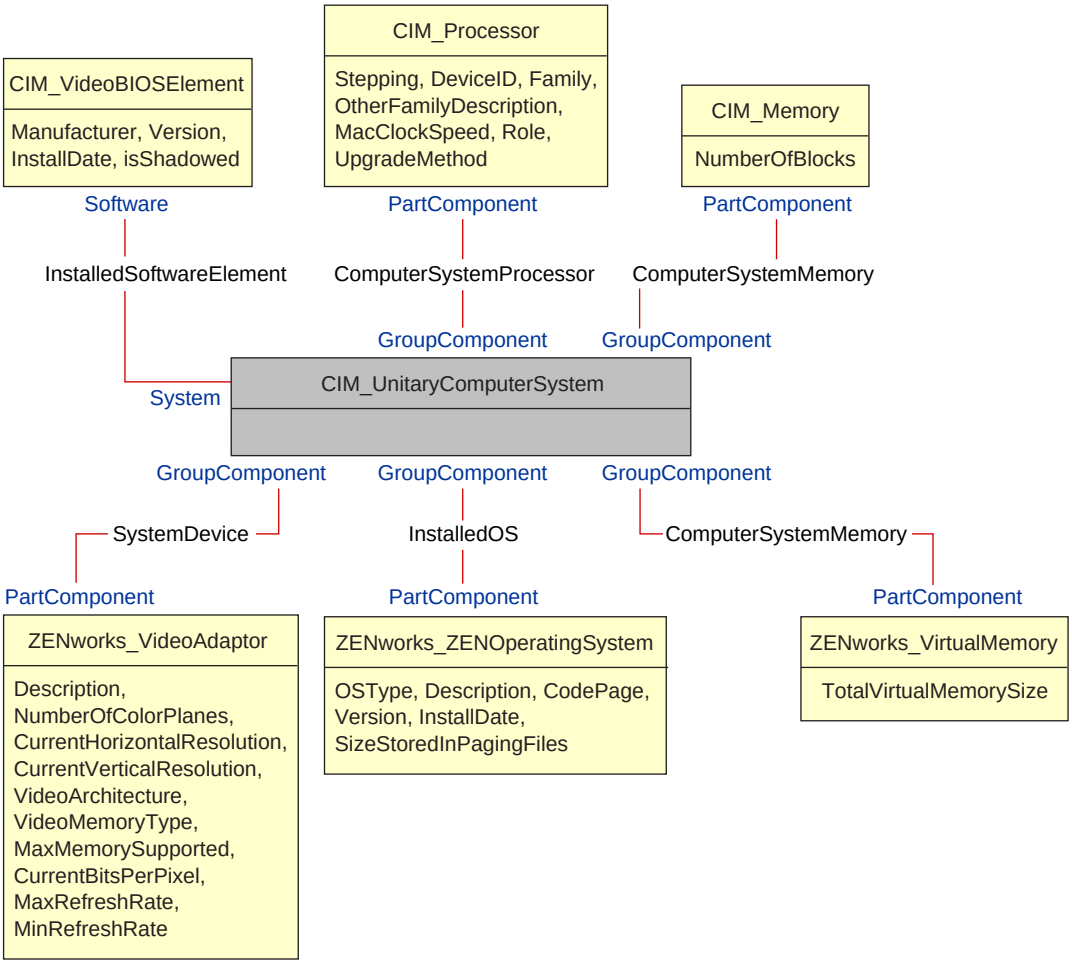
The elements of the meta schema are classes, properties, and methods. The meta schema also supports indications and associations as types of classes and references as types of properties.

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.

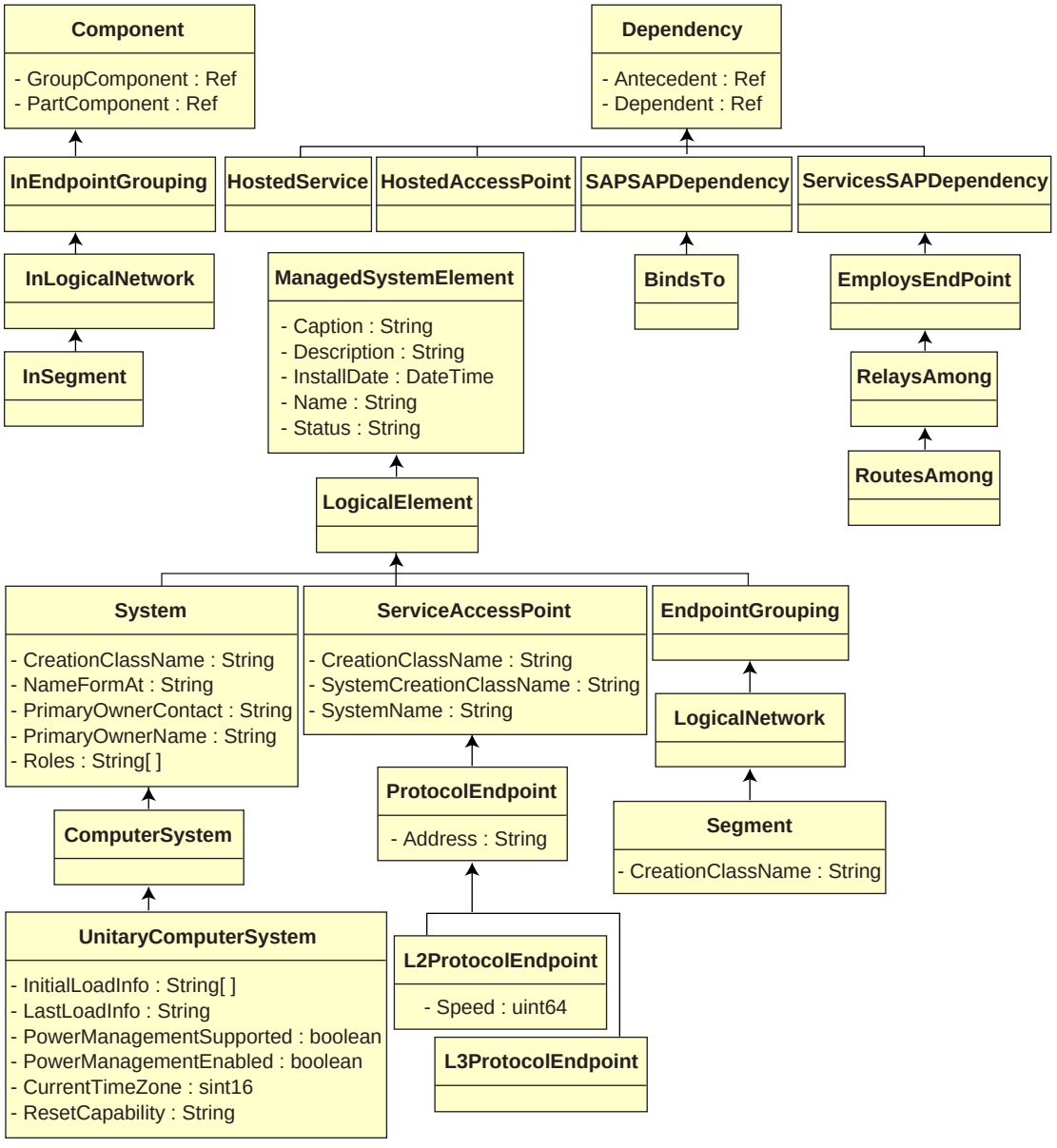
An association is a special class that contains two or more references. It represents a relationship between two or more objects. Because of the way associations are defined, it is possible to establish a relationship between classes without affecting any of the related classes. That is, addition of 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 ZfD uses.



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

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



CIM-to-Relational Mapping

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

- **Logical Schema:** The logical schema defines how the 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.

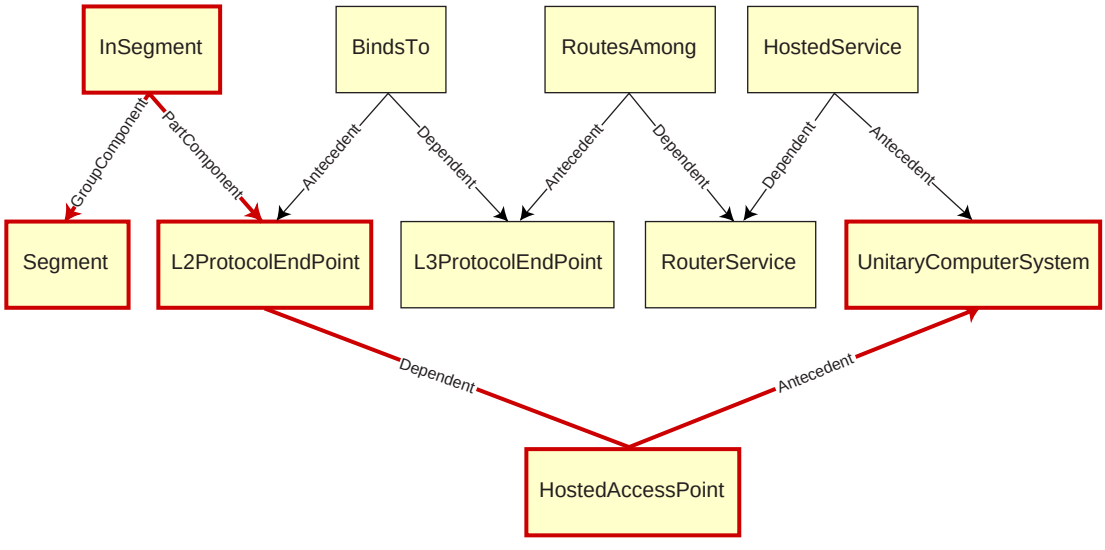
A table in the database represents each class in the CIM hierarchy. A column of the appropriate type in the table represents each non-inherited property in the class. Each table also has a primary key, id$, which is a 64-bit integer that uniquely identifies an instance. An instance of a CIM class is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

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 349
- “Physical Schema” on page 355

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 353.
- A destructor procedure to destroy an instance of the class. For more information, see "Destructor" on page 354.
- A view to access and update the values of properties of the class.

ZfD 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

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 ZfD 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 ZfD, the Inventory Query component is not affected because the queried data is retrieved from the database before being queried and so case sensitivity is automatically taken care of.

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

Views

Each CIM class is represented in the database by a view that contains all the local and inherited non-array properties of that class. The view is named the same as the CIM class. 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 352.
- **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 pid, 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 pid. 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 ZfD

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

- “Case Study of CIM Schema Implementation in ZfD” on page 355
- “Legends for Schema Diagrams” on page 358
- “Leaf Objects in the Database” on page 358
- “Non-CIM Tables and Views in the ZfD Database Schema” on page 359
- “CIM Classes and Extension Classes in ZfD” on page 359
- “Schema Diagrams of CIM and the Extension Schema in ZfD” on page 361
- “Sample Inventory Database Queries” on page 368

Case Study of CIM Schema Implementation in ZfD

The following scenario describes a managed workstation 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.

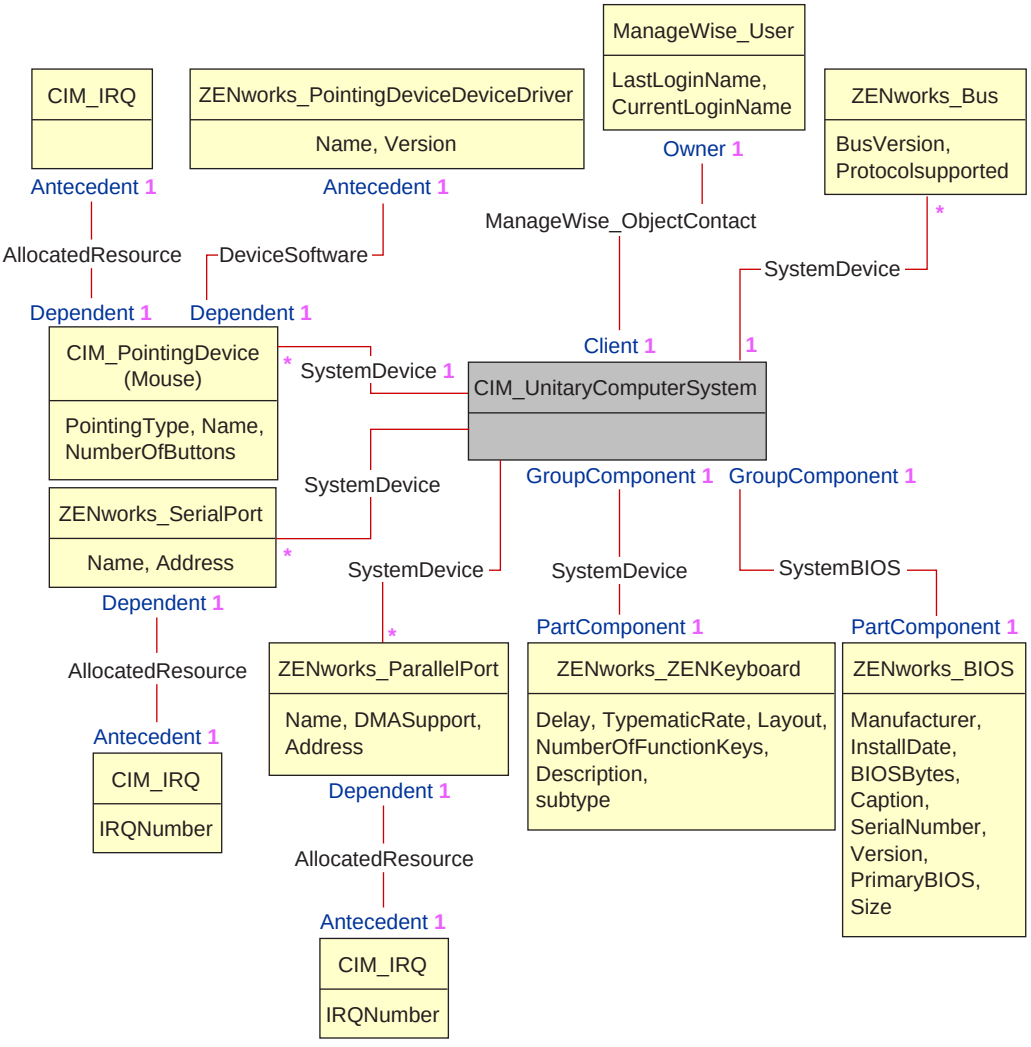
Class ZENworks.BIOS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBIOS.PartComponent pointing to ZENworks.BIOS. The relationship between the two classes is one to one. This means a computer 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.User associates to CIM.UnitaryComputerSystem using the association CIM.ManageWise.ObjectContact with Client.

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 ZfD” on page 359. For schema diagrams of other classes, see “Schema Diagrams of CIM and the Extension Schema in ZfD” on page 361.

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.
- 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).

Leaf Objects in the Database

A leaf object in the Inventory database has only one instance at any given point of time. In terms of relational mapping, a table that is made as a leaf object will have only one row and this row will be shared across multiple workstation objects in the database. The advantages of this approach are better optimized storage and better performance for queries.

Any Inventory component object behaves like a leaf object in the database if it is same across several workstations and no other objects are associated under it. For example, a software application installed on the workstation is a leaf object. The following classes are leaf nodes in the ZfD Inventory database:

1. ZENworks_PointingDeviceDeviceDriver
2. ZENworks_ZENKeyboard
3. ZENworks_Bus
4. CIM_IRQ
5. CIM_DMA
6. ZENworks_ZENOperatingSystem
7. CIM_Processor
8. CIM_Memory
9. CIM_VideoBIOSElement
10. CIM_Product
11. ZENworks_NetwareClient
12. CIM_POTSModem
13. ZENworks_SoundCard
14. ZENworks_MotherBoard
15. CIM_PowerSupply
16. ZENworks_PhysicalDisk
17. ZENworks_PhysicalDiskette
18. ZENworks_LogicalDiskette
19. ZENworks_PhysicalCDROM
20. Zenworks.NetworkAdapterDriver

"CIM Classes and Extension Classes in ZfD" on page 359 describes the CIM classes that ZfD uses.

Non-CIM Tables and Views in the ZfD Database Schema

There are a few tables called "support tables" in the ZfD database. These tables and views do not follow the CIM specification. These tables and views are as follows:

View Name	Description	Table that the View Uses	Inventory Component that Uses the View
ZENworks.Processor.Family	Contains the enum values and strings indicating the processor families. These enum values are described in the DMI specification.	t$ProcessorFamily	Inventory Reporting
ZENworks.ProcessorRole	Contains the enum values and strings indicating the processor roles. These enum values are described in the DMI specification.	t$ProcessorRole	Inventory Reporting
ZENworks.OperatingSystemType	Contains the enum values and strings indicating the types of operating systems. These enum values are described in the DMI specification.	t$OperatingSystemType	Inventory Reporting
ZENworks.VideoArchitecture	Contains the enum values and strings indicating the different video adapter types. These enum values are described in the DMI specification.	t$VideoArchitecture	Inventory Reporting
MW_DBA.LockTable	Inventory Storer maintains modification time stamps indicating the last modified time of the inventory information. The Storer will not store the inventory scan received if the time stamp is the same.	t$LockTable	Inventory Storer

CIM Classes and Extension Classes in ZfD

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

CIM and Extension Class in ZfD	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.Site	Site ID and site name of the Inventory database.
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.

CIM and Extension Class in ZfD	Description of the details that the Class Models
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 workstation.
ManageWise.MSDomainName	Name of the domain to which the Windows NT workstation is attached.
ManageWise.NDSName	DN name and tree under which the managed workstation is registered in eDirectory.
CIM.VideoBIOSElement:	Video driver.
CIM.Processor	Processor of the workstation.
CIM.Memory	Total memory of the workstation.
ZENworks.VirtualMemory	Total virtual memory size of the workstation.
ZENworks.Videoadapter	Properties of the monitor and the adapter connecting it.
ZENworks.ZENOperatingSystem	Details of the operating system.
ZENworks.InventoryScanner	Details of the inventory scanner that has scanned for hardware and software details of the managed workstation
ZENworks.NetwareClient	NetWare client version of the workstation
CIM.Product	Software installed on the managed system. Key attributes are the names of the product, vendor, and version.
CIM.EthernetAdapter	Information on the properties of the network adapter.
ZENworks.NetworkAdapterDriver	Network card adapter driver information.
CIM.IPProtocolEndpoint	IP address of the workstation.
CIM.IPXProtocolEndpoint	IPX address of the workstation.
CIM.LANEndpoint	Active MAC address.
ManageWise.DNSName	DNS name of the workstation.
ZENworks.SoundCard	Description of the multimedia adapter on the workstation.
CIM_POTSModem	Physical configuration of the modem device.
CIM_DMA	Information about the system DMA channels.
CIM.CacheMemory	Information about the configured system cache.

CIM and Extension Class in ZfD	Description of the details that the Class Models
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 workstation.
CIM.PowerSupply	Information about the power supply unit of the workstation.
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 workstation, 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.LocalFileSystem	Information about the local file system mounted on the hard disk.
ZENworks.PhysicalCDROM	Derived from ZENworks.StoragePhysicalMedia to model the CD drive.
ZENworks.LogicalCDROM	Drive mapped to the CD drive.
ZENworks.BackupDisk	Derived from ZENworks.StoragePhysicalMedia to model other backup disks like Jaz drives and Zip disks.
ZENworks.LogicalBackupDisk	Drives mapped to the backup devices on the workstation.

Schema Diagrams of CIM and the Extension Schema in ZfD

The following schema diagrams of the CIM and extension schema model the Inventory database in ZfD:

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.

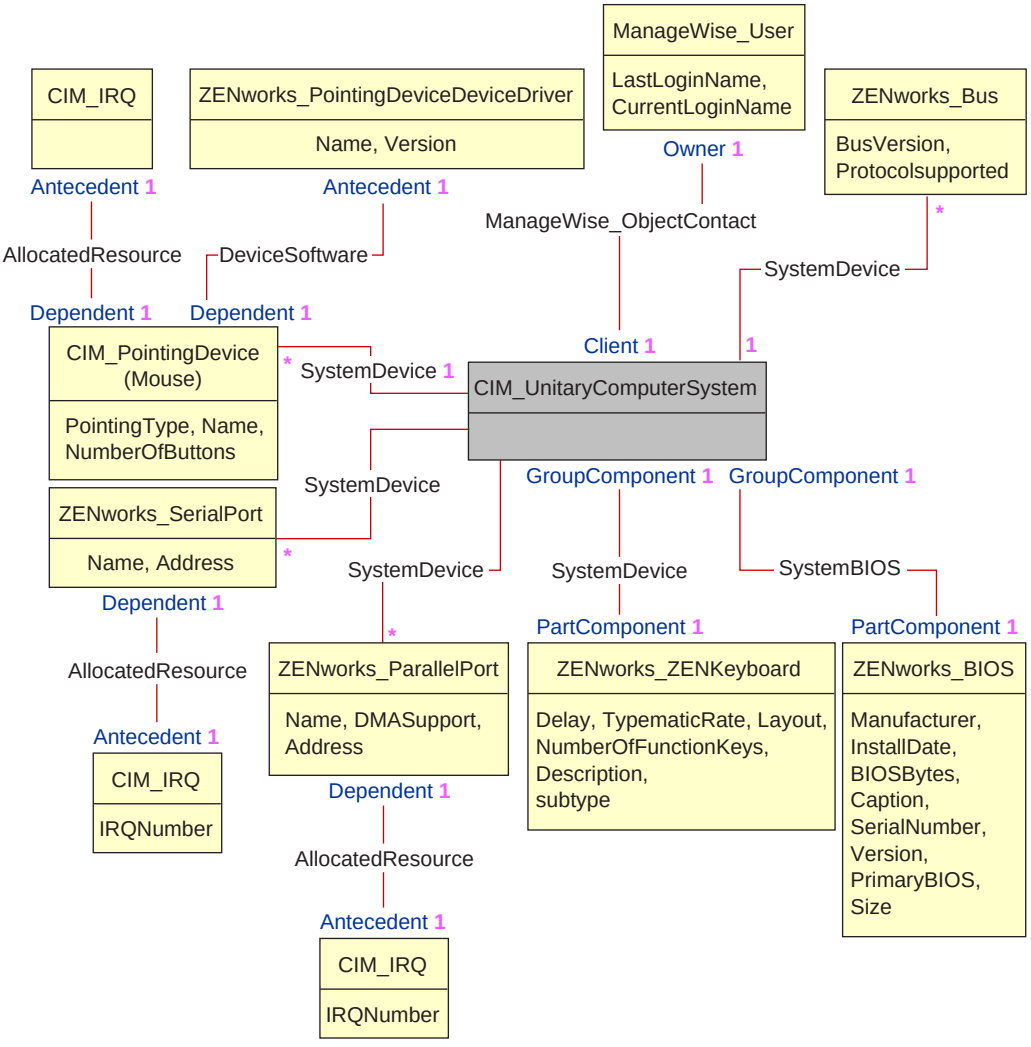
Class ZENworks.BIOS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBIOS.PartComponent pointing to ZENworks.BIOS. The relationship between the two classes is one to one. This means a computer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

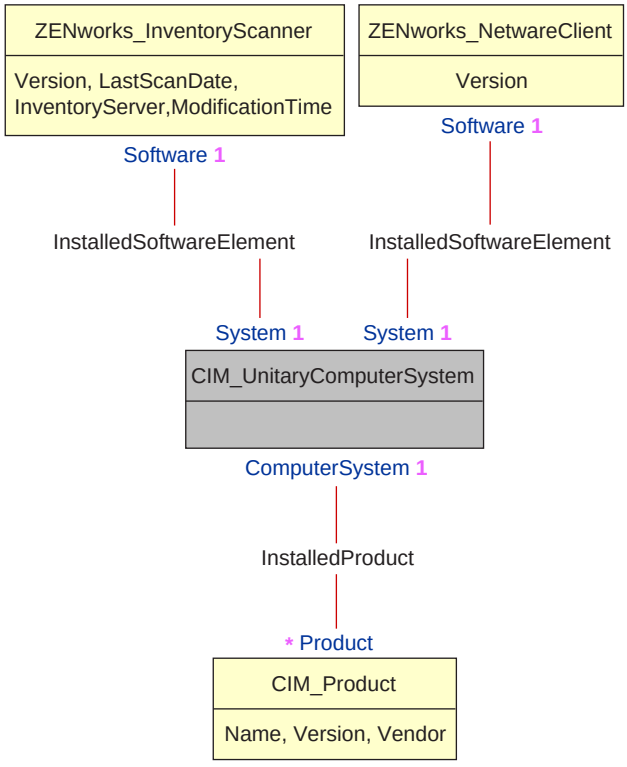
Class CIM.ZENworks.ParallelPort associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemDevice.PartComponent pointing to CIM.ZENworks.ParallelPort. The relationship between the two classes is one to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

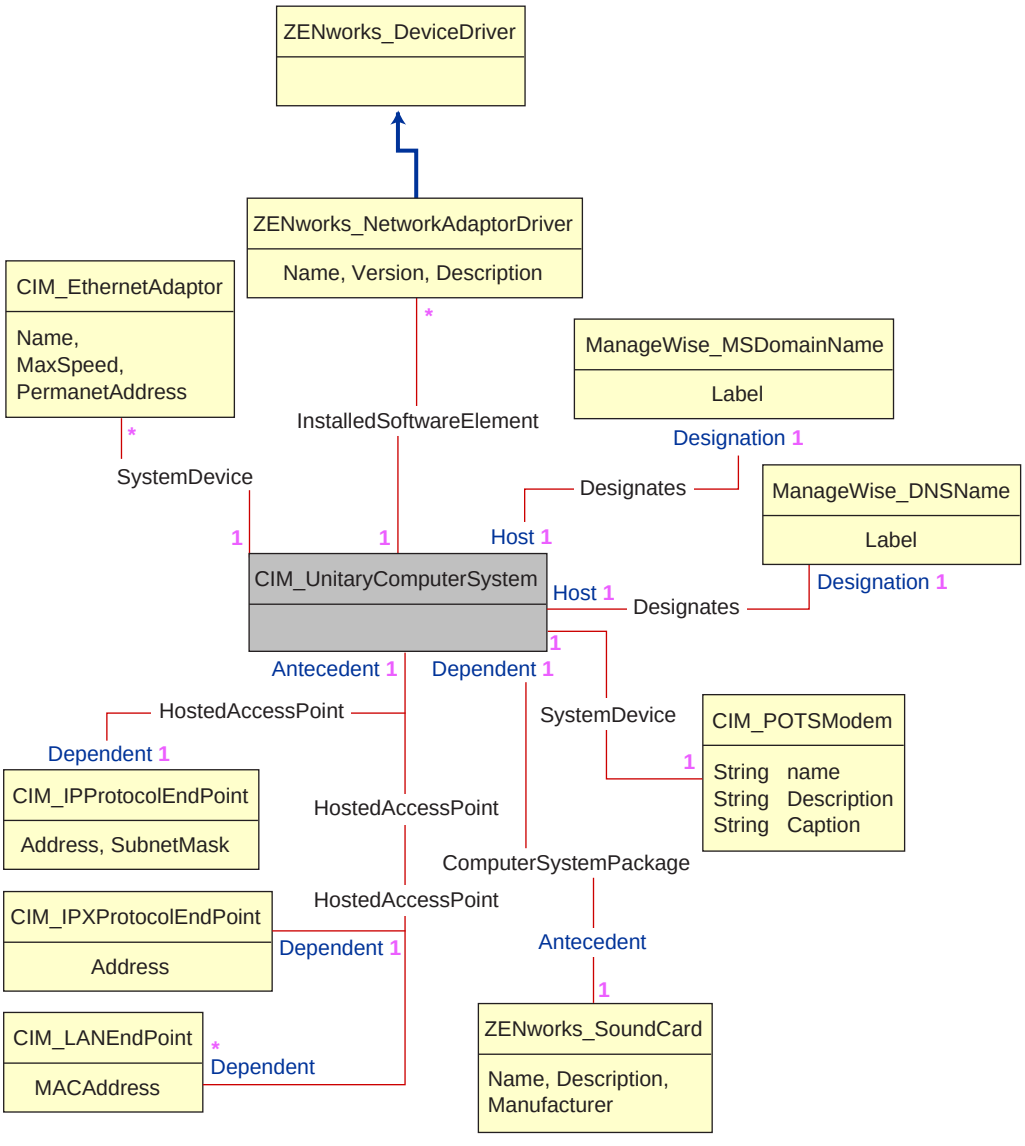
Class ZENworks.BUS associates to CIM.UnitaryComputerSystem using the association CIM.SystemDevice with SystemDevice.GroupComponent pointing to CIM.UnitaryComputerSystem and SystemBUS.PartComponent pointing to ZENworks.BUS. The relationship between the two classes is one to one. This means a computer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

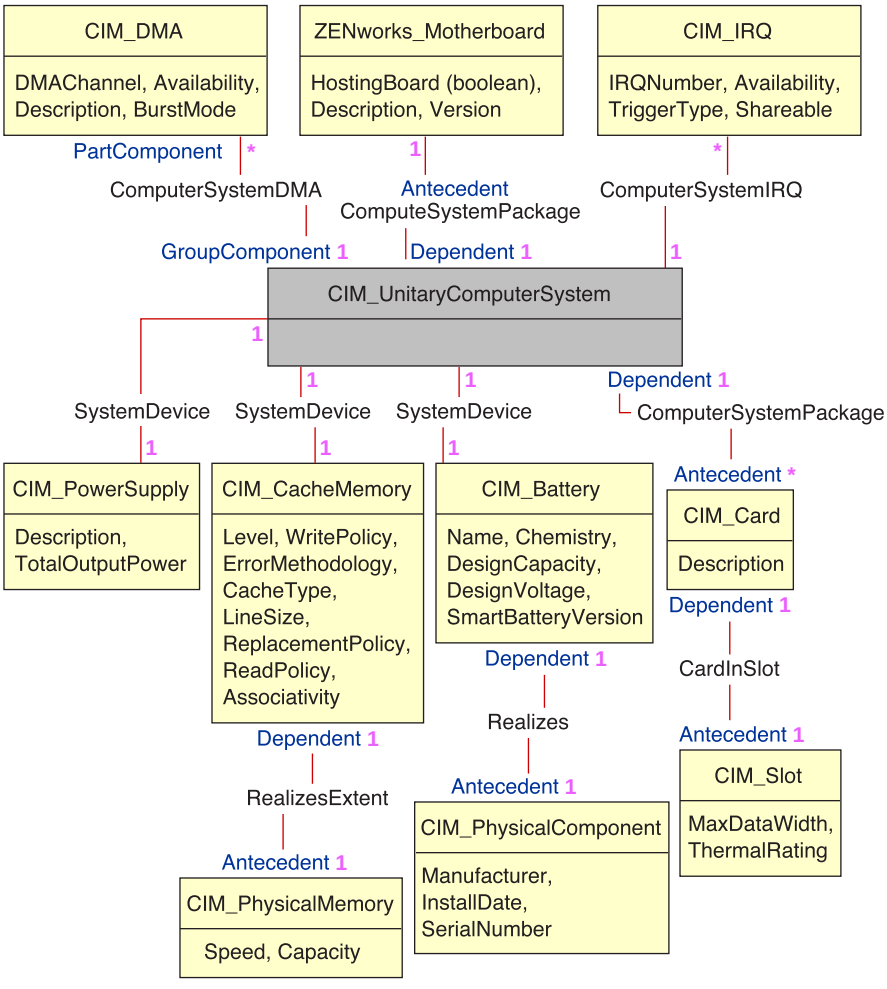
Class ManageWise.User associates to CIM.UnitaryComputerSystem using the association CIM.ManageWise.ObjectContact with Client.

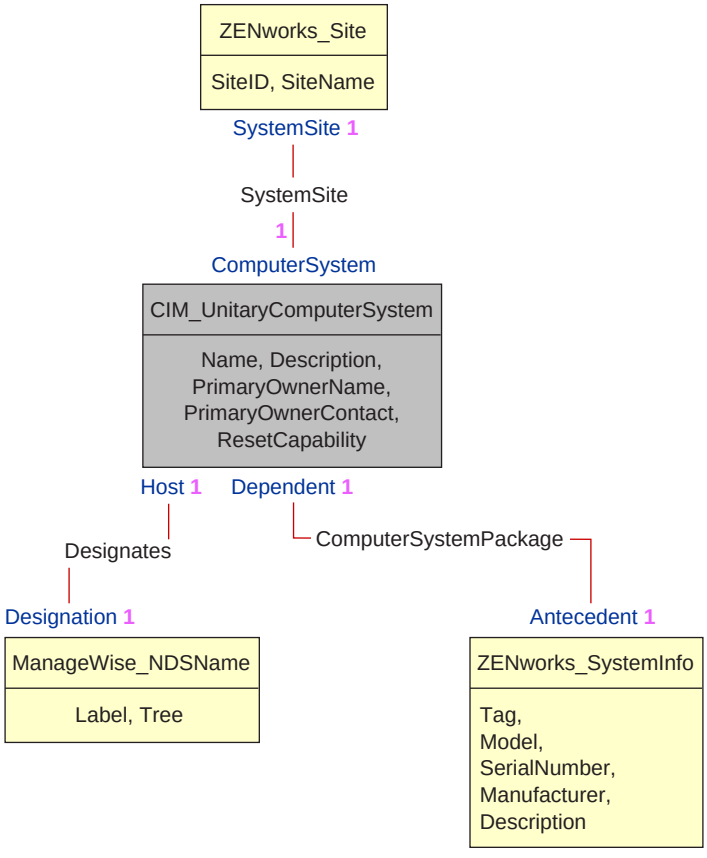
Class CIM.IRQ associates to CIM.ParallelPort using the association CIM.AllocatedResource. Dependent pointing to CIM.ParallelPort and Antecedent pointing to CIM.IRQ.

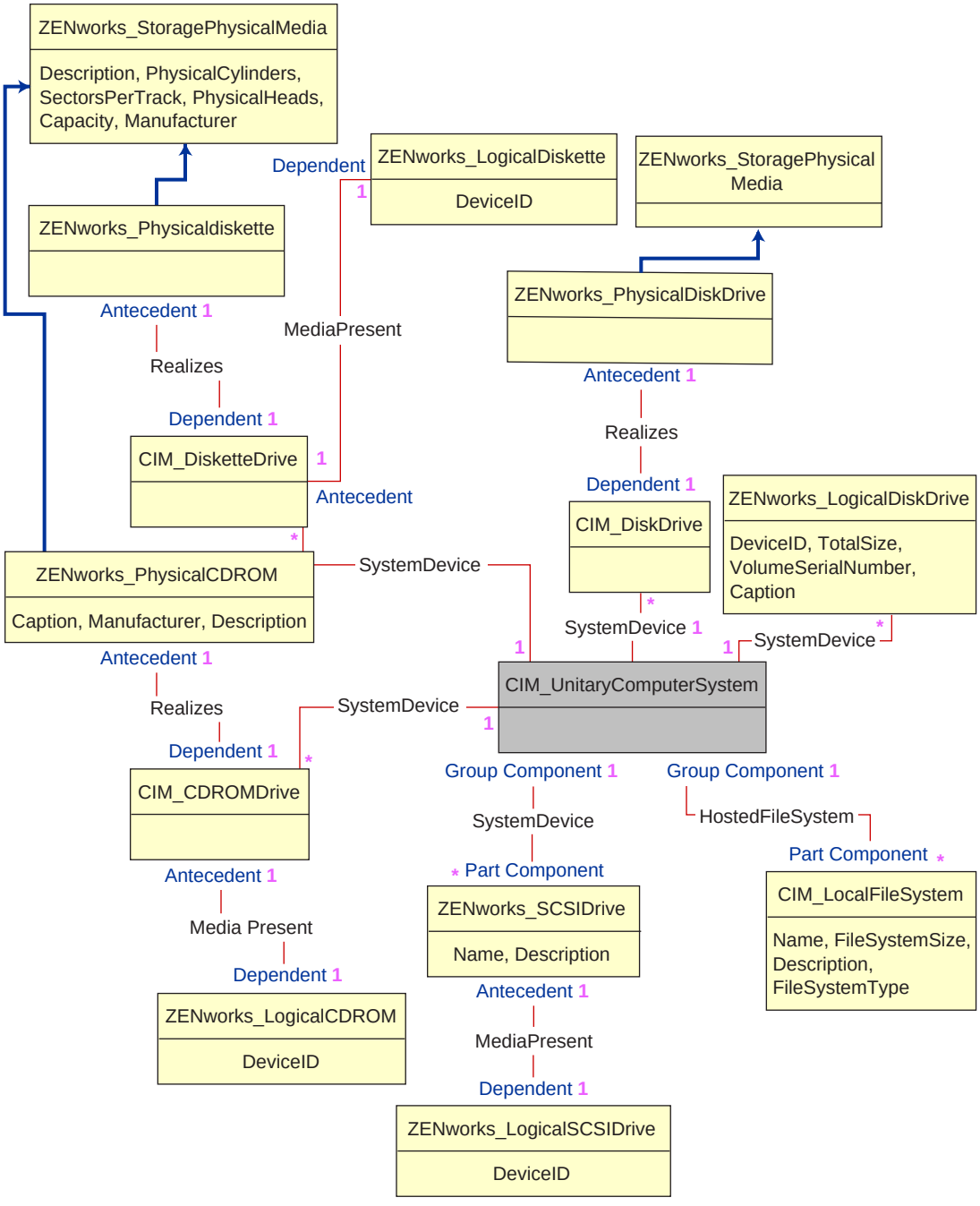

ManageWise_User
LastLoginName, CurrentLoginName
CIM_IRQ
ZENworks_PointingDeviceDeviceDriver
Name, Version
ZENworks_Bus
BusVersion, Protocolsupported
Owner 1
Antecedent 1
Antecedent 1
ManageWise_ObjectContact
AllocatedResource
DeviceSoftware
SystemDevice
*
Dependent 1
Dependent 1
Client 1
1
CIM_PointingDevice (Mouse)
PointingType, Name, NumberOfButtons
*
SystemDevice 1
CIM_UnitaryComputerSystem
SystemDevice
GroupComponent 1
GroupComponent 1
ZENworks_SerialPort
Name, Address
*
SystemDevice
SystemDevice
SystemBIOS
Dependent 1
*
PartComponent 1
PartComponent 1
AllocatedResource
ZENworks_ParallelPort
Name, DMASupport, Address
ZENworks_ZENKeyboard
Delay, TypematicRate, Layout, NumberOfFunctionKeys, Description, subtype
ZENworks_BIOS
Manufacturer, InstallDate, BIOSBytes, Caption, SerialNumber, Version, PrimaryBIOS, Size
Antecedent 1
CIM_IRQ
IRQNumber
Dependent 1
AllocatedResource
Antecedent 1
CIM_IRQ
IRQNumber


ZENworks_InventoryScanner
Version, LastScanDate, InventoryServer,ModificationTime
ZENworks_NetwareClient
Version
Software 1
Software 1
InstalledSoftwareElement
InstalledSoftwareElement
System 1
System 1
CIM_UnitaryComputerSystem
ComputerSystem 1
InstalledProduct
* Product
CIM_Product
Name, Version, Vendor


ZENworks_DeviceDriver
ZENworks_NetworkAdaptorDriver
Name, Version, Description
CIM_EthernetAdaptor
Name,
MaxSpeed,
PermanetAddress
ManageWise_MSDomainName
Label
InstalledSoftwareElement
Designation 1
SystemDevice
Designates
ManageWise_DNSName
Label
Host 1
CIM_UnitaryComputerSystem
Designation 1
Host 1
Designates
Antecedent 1
Dependent 1
HostedAccessPoint
SystemDevice
CIM_POTSModem
String name
String Description
String Caption
Dependent 1
CIM_IPProtocolEndPoint
Address, SubnetMask
HostedAccessPoint
ComputerSystemPackage
HostedAccessPoint
CIM_IPXProtocolEndPoint
Address
Dependent 1
Antecedent
CIM_LANEndPoint
MACAddress
Dependent
ZENworks_SoundCard
Name, Description,
Manufacturer


CIM_DMA
DMAChannel, Availability, Description, BurstMode
ZENworks_Motherboard
HostingBoard (boolean), Description, Version
CIM_IRQ
IRQNumber, Availability, TriggerType, Shareable
PartComponent
*
ComputerSystemDMA
1
Antecedent
ComputeSystemPackage
*
ComputerSystemIRQ
GroupComponent 1
Dependent 1
1
CIM_UnitaryComputerSystem
1
1
1
1
Dependent 1
SystemDevice
SystemDevice
SystemDevice
ComputerSystemPackage
1
1
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
Antecedent 1
CIM_Slot
MaxDataWidth, ThermalRating
RealizesExtent
Antecedent 1
CIM_PhysicalComponent
Manufacturer, InstallDate, SerialNumber
Antecedent 1
CIM_PhysicalMemory
Speed, Capacity


ZENworks_Site
SiteID, SiteName
SystemSite 1
SystemSite
1
ComputerSystem
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



Sample Inventory Database Queries

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

Refer to the schema diagrams in "Schema Diagrams of CIM and the Extension Schema in ZfD" on page 361 to find out the associated schema classes and attribute information.

1. Retrieve the name and ID of all workstations from the database and also to the tree to which these workstations 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 workstations 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 workstation 'SJOHN164_99_139_79.WS' registered under the 'NOVELL_AUS' 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 workstation 'SJOHN164_99_139_79.WS'. The query is as follows:

```
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
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 workstation 'SJOHN164_99_139_79.WS'. The query is as follows:

```
SELECT s.host FROM managewise.NDSName m,managewise.Designates s WHERE
m.label='SJOHN164_99_139_79.WS' and m.id$=s.Designation
```

6. When you know the ID of the UnitaryComputerSystem for a particular workstation 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 workstation in place of the ? value for u.id in the query.

7. List the IP address, IPX address, and MAC address of all workstations 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 workstation 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 workstation 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 workstation.

```
SELECT m.DEVICEID, m.TotalSize, m.VolumeSerialNumber, m.Caption FROM
zenworks.LogicalDiskDrive m, cim.UnitaryComputerSystem u,
cim.SystemDevice s WHERE s.PartComponent=m.id$ AND s.GroupComponent=u.id$
and u.id$=?
```

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

25 Setting Up Workstation Inventory

The following sections contain detailed information to help you set up Novell® ZENworks® for Desktops (ZfD) Workstation Inventory:

Configuring the Settings for the Inventory Service Object

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

- Inventory Server Role
- Discard Scan Data Files
- Scan Directory Path
- Enable scan of workstations
- Start full scan

To open the Inventory Service Object properties page:

1. In ConsoleOne®, right-click the Inventory Service object (*servername*_ZenInvservice) > click Properties > click the Inventory Service Object Properties tab.
2. Modify the following settings:

Inventory Server Role: Based on the servers that you have deployed for scanning inventory, you must specify the role of the server. See "Changing the Role of the Inventory Service Object" on page 381.

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 server, which is one of the following types: Intermediate Server, Intermediate Server with Database, Intermediate Server with Database & Workstations, and Intermediate Server with Workstations.

Scan Directory Path: When you install ZfD, you specify the volume on the server for storing the scan data files. If required, you can modify 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 server that stores the scan data files. The format of the Scan Directory Path is as follows: *server_name\volume_of_the_server_directory*.

You cannot modify the server name specified in the SCANDIR path. If you modify the directory, the directory must already exist.

To modify the path, click the Browse button.

Enable Scan of Workstations: To scan the workstations 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 workstations, uncheck this option.

Start Full Scan: When scanning the workstation for the first time, the Scanner collects the complete inventory of the workstation. A complete inventory scan of the workstation is referred as a *full scan.* After the workstation is scanned, the next time the Scanner compares the current inventory data to the history data that it maintains. If there are any changes to the workstation, the Scanner creates a *delta scan*, which collects the changes in inventory since the last scan was done. The delta scan setting is the default scan operation for each successive scan after the first scanning of the workstation. If the Status Log reported by the inventory component indicates the scanning on the workstation is not successful, you can enforce a full scan. This policy settings is applicable for all workstations associated with it. To override this policy, you set this option for an individual workstation. For more information about the Workstation object settings, see "Configuring the Inventory Settings for the Workstation Object" on page 372 the Workstation.

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.

Configuring the Inventory Settings for the Workstation Object

You use the inventory settings of the Workstation object to identify the .STR filename of the last scan. Identifying the scan data filename is useful to troubleshoot inventory scanning problems for the workstation. For more information, see View the Scan Data File.

You configure the inventory settings for the selected workstation by enforcing a complete scan of the workstation using the Workstation Scan Configuration property page. For more information about the Full Scan option, see Enable Full Scan.

To configure the inventory settings for the workstation object:

1 In ConsoleOne, right-click the Workstation object > click Properties > ZENworks Inventory > Workstation Scan Configuration properties tab.

2 Check Start Full Scan if required > modify the following settings:

View the Scan Data Filename: The Scanner stores the scan data of the workstation as a scan data file (.STR) in the scan directory (SCANDIR) on the inventory server. The Workstation Scan Configuration properties page displays the scan filename of the last scan done at the workstation. The .STR file uses the following filename conventions: *macAddress_gmt_sequencenumber*.STR where *macAddress* is the MAC address of the workstation, *gmt* is the time that the workstation was first scanned, *sequencenumber* is the internal sequencing number generated by the Scanner for the workstation while scanning, and .STR is the file extension.

For example, 00508b12b2c4_944029836000_10.STR is the .STR file for the workstation with the MAC address of 00508b12b2c4, the GMT of the workstation as 944029836000, and the internal sequencing number of 10.

Start Full Scan: When scanning the workstation for the first time, the scanner collects the complete inventory. A complete inventory scan of the workstation is referred to as a *full scan*. After the workstation is scanned, the next time the scanner compares the current inventory data to the history data that it maintains. If there are any changes to the workstation, the scanner creates a *delta scan*, which collects the changes in inventory since the last scan was done. The delta scan setting is the default scan operation for each successive scan after the first scanning of the workstation. If the status log reported by the inventory component indicates that the scanning on the workstation was not successful, you can enforce a full scan.

3 Click OK.

Configuring the Workstation Inventory Policy Settings

You can set policies to control how workstations collect inventory. The Inventory policy settings configure the inventory scanning options for the selected Workstation Inventory Package. The Inventory policy settings stored in Novell eDirectory™ are associated with a Workstation object. Each Workstation object has an associated Inventory policy package.

From the Inventory policy, you can configure the following:

- DN of the Inventory Service Object
- Enable Software Scan
- Software Applications in the Custom Scan Editor

To set a Workstation Inventory policy:

1. Create or modify a workstation policy package.
2. Enable the Workstation Inventory policy.
3. Set the details in the Workstation Inventory policy.

You can complete these steps on your own or use the Policy Wizard to guide you through the process.

To configure the Workstation Inventory policy:

1 In ConsoleOne, right-click the Workstation Inventory policy package > click Properties > click Policies > click one of the following subtabs: Win95-98 or WinNT-2000.

2 Click the Workstation Inventory row > Properties > Workstation Inventory Policy tab.

3 Modify the settings.

Inventory Service Object: The installation program creates the Inventory Service object and copies the inventory components on the server. In this policy, you choose an Inventory Service object to which the scanners will send the inventory information of the workstations associated with this server.

Enable Software Scan: By default, the scanners collect hardware information of the workstations. When you check the Enable Software Scan option, the next time the workstations associated with this policy package are scanned, the scanners will collect software information. For more information, see "Software Information Collected by the Scanners" on page 297.

Custom Scan Editor: If you enable the software scanning of the workstations, specify the list of software applications that will be scanned for at the workstation in the Custom Scan Editor. For more information, see "Customizing Software Scanning of Workstations" on page 378.

4 Click OK.

NOTE: If you are modifying the Inventory policies or configuring the objects, always stop the Inventory Services. Configure the policies and properties of the objects. Restart the Inventory services again.

Configuring the Roll-Up Policy Settings

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

To set up the Roll-Up policy:

1. Create or modify a server package.
2. Enable the Roll-Up policy.
3. Configure the Roll-Up policy.

To open the Roll-Up policy:

1 In ConsoleOne, right-click the Server package > click Properties > Policies > click one of the following subtabs: General, NetWare, WinNT-2000.

2 Check the check box under the Enabled column for the zeninvRollup Policy.

3 Click Properties.

4 Modify the setting.

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

NOTE: Ensure that the specified server is a different server because the roll-up of data cannot happen on the same server. Also, the higher level roll-up server cannot specify the lower-level server as the next-destination server for roll-up of data.

5 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.

Scanning for IBM Computer Models

The Scanner (WINSCAN.EXE) uses the IBMNAMES.INI file to get information about IBM* workstations running under Windows* 95/98. The scanner reads the model name using the machine type and model number information specified in the IBMNAMES.INI file. This file is located in the SYS:\PUBLIC\ZENWORKS directory of the inventory server.

Using the IBMNAMES.INI file is an IBM-specific method of scanning model names of IBM workstations defined by the scanner. If you have a new IBM workstation that is not listed in the IBMNAMES.INI file, the model number of the workstation will not be scanned. You must edit the IBMNAMES.INI file to add the machine type, model number, and model description of the new IBM workstation. By adding this entry, you enable the scanner to identify the new model name

Editing the IBMNAMES.INI File to Add the IBM Computer Models

The IBMNAMES.INI file contains the model number and description of IBM desktops and laptop models. You add entries in this file to specify the models that the scanner should scan for at the workstations.

This file begins with a section called [Product Names].

The format of each entry in the section is as follows:

`4_bytes_machine_type-3_byte_model_number=`*`model_description`*

For example, if the model is IBM PC 140 and the machine type is 6260, specify the model description as IBM PC 140. The entry in the IBMNAMES.INI is 6260-79T = IBM PC 140.

If you want the scanner to scan for all IBM computer models of a particular machine type with the same model description, the *3_byte_model_number* is specified with three question marks (???) as wildcard characters.

A sample IBMNAMES.INI file is as follows:

```
[Product Names]

6272-???=IBM PC 300GL (6272)

6260-???=IBM PC 140 (6260)

6282-???=IBM PC 300Gl (6282)

2460-79T=IBM ThinkPad 5602 (2641)

...
```

For example, to scan all models of a 6282 machine type with same model description, the file entry is as follows:

```
6282-???=IBM PC 300GL (6282)
```

The machine type and model number are printed at the rear of the laptop or on the backside of the desktop workstation. For example, the 760E Thinkpad* model has the following label: TYPE *9546-A98*.

Customizing the Hardware Scanning Information of Jaz, Zip, and Floppy Drive Vendors

The scan information of the vendors for devices such as backup and floppy devices is usually unavailable on the workstation. 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 the Hardware Rules file. The scanners read this file during the hardware scanning process for these devices.

The contents of the Hardware Rules file (ZIPPNAMES.INI) is as follows:

```
[Identifier]

device_id=vendor_display_name_you_specify

[Floppy]

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 workstation.

For example, the contents of the file are as follows:

```
[Identifier]

IOMEGA ZIP 100 D.13=Iomega Corporation
```

This entry is for a 100 MB Zip* drive installed on the workstation.

```
[Floppy]

MITBISHI LS-120 F200 08=MITBISHI
```

This entry is for a LS-120 floppy drive installed on the workstation.

To customize and update the vendor information for display:

1. Open the ZIPPNAMES.INI file in a text editor.

   This text file is located in \PUBLIC\ZENWORKS directory on the inventory server.

2. Add or modify the entries in this file.

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

3. Save the changes.

Scanning for Vendor-Specific Asset Information from DMI

Follow these steps:

1. Modify the asset file, ASSET.INI. For more information, see “Asset File Format” on page 377.

2. Run the scans on the workstations.

   Verify that the inventory information is in eDirectory Minimal Information and the Workstation Summary window.

Asset File Format

The asset file, ASSET.INI contains the following three sections:

- Contains Serial Number in the section [SERIALNUMBER]
- Contains Asset Tag in the section [ASSETTAG]
- 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. These three classes contain the Asset Information in DMI.

The format of the ASSET.INI file is as follows:

```
[SERIALNUMBER]

DMI1_CLASSNAME=DMI_class_pathname_for_serial_number

DMI1_ATTRIBUTEID=DMI_attribute_ID_for_serial_number

[ASSETTAG]

DMI1_CLASSNAME=DMI_class_pathname_for_asset_tag

DMI1_ATTRIBUTEID=DMI_attribute_ID_for_asset_tag

[MODEL]

DMI1_CLASSNAME=DMI_class_pathname_for_computer_model

DMI1_ATTRIBUTEID=DMI_attribute_ID_for_computer_model
```

The SerialNumber and Model values can have string lengths of maximum 64 characters. The Assettag value can have string length of maximum 256 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.

A sample ASSET.INI file configuration is as follows:

```
[SERIALNUMBER]

DMI1_CLASSNAME=IBMPSG|Serial Number Information|001

DMI1_ATTRIBUTEID=9

DMI2_CLASSNAME=IntelG|Serial Number Information|002

DMI2_ATTRIBUTEID=8

[ASSETTAG]

DMI1_CLASSNAME=Novell|AssetInformation|001

DMI1_ATTRIBUTEID=2

DMI2_CLASSNAME=IntelG|Asset Information|001

DMI2_ATTRIBUTEID=3
```

```
[MODEL]

DMI1_CLASSNAME=Novell|SystemModel|001

DMI1_ATTRIBUTEID=2

DMI2_CLASSNAME=IntelG|SystemModel|002

DMI2_ATTRIBUTEID=4

[COMPUTERTYPE]

DMI1_CLASSNAME=Novell|ComputerType|001

DMI1_ATTRIBUTEID=1

[MODELNUMBER]

DMI1_CLASSNAME=Novell|ComputerModelNumber|001

DMI1_ATTRIBUTEID=1
```

Customizing Software Scanning of Workstations

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

By default, the Scanner will not scan for software applications at the workstation. You must enable the Software Scan option in the Workstation Inventory policy.

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 workstation scanning:

- “Adding New Applications for Scanning” on page 378
- “Format of the Custom Scan File” on page 379
- “Exporting the List of Application Files for Scanning” on page 380

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 Workstation Inventory policy.

   For more information, see “Configuring the Workstation Inventory Policy Settings” on page 373.

   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, File name, 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 379.

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 379.
3. Save the application as a text file with any extension you prefer.
4. In ConsoleOne, open the Workstation 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 Workstation Inventory policy.

   For more information, see "Configuring the Workstation Inventory Policy Settings" on page 373.

   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.

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 file. The scanners read the file during the scanning process for displaying the vendor or product name as specified in the file.

The contents of the Software Rules file are 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

To customize the vendor and product names for display:

1. Open the SWRULES.INI file in a text editor.

The SWRULES.INI file is a text file located in \PUBLIC\ZENWORKS on the inventory server.

2. 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.
   ```

3. 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®
   ```

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

   ```
   [vendor]

   others=null
   ```

You should follow these rules while editing the SWRULES.INI file:

- Ensure that blank lines do not exist between the sections of the file.
- The file 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.

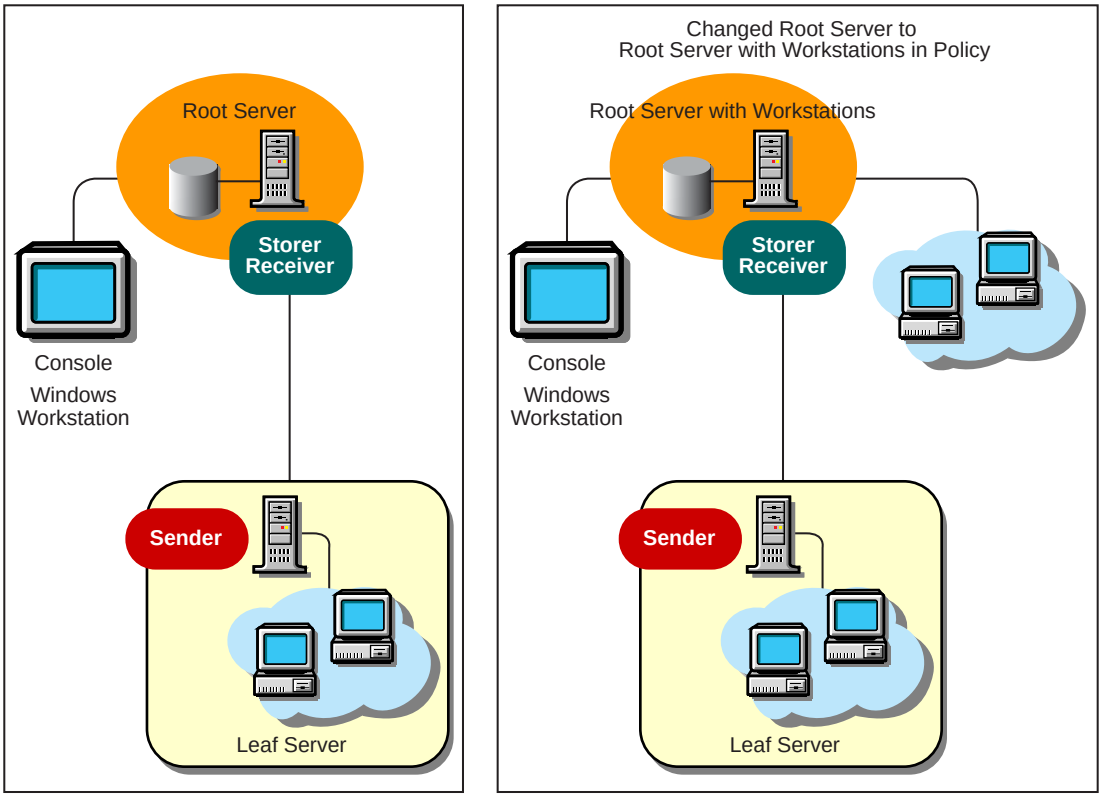
If you specify incorrect entries in the file, the entries preceding the incorrect entry will be used and the other entries will be ignored.

Changing the Role of the Inventory Service Object

When you install ZfD, you assign specific roles to the server based on your inventory deployment. For example, if the deployment plan identifies three servers, such as a Root Server, an Intermediate Server with Database, and a Leaf Server for inventory deployment, you install Workstation Inventory on these servers, and choose the role for the server. Later, if you want to make changes

in the inventory deployment, such as attaching the workstations to the existing Root Server, you need to change the role of the Inventory Service object from Root Server to Root Server with Workstations. Additionally, depending on the new role, there are some policies you need to configure.

The following illustration shows the current deployment and the changes in the deployment:



Existing Inventory Deployment

Changes in the Inventory Deployment Through Policies

To change the role for any 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. For more information, see Workstation Inventory in *Deployment*.
2. In ConsoleOne, right-click the Inventory Service object (*servername*_ZenInvservice) > click Properties > click the Inventory Service Object Properties tab.
3. Choose the new role of the Inventory Service object > 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 Workstations, you need to configure the Workstation Inventory policy for the workstations that you have attached. Similarly, to change the role to any other server, you need to follow the instructions to make the new role change effective.
4. Bring down the services running on the changed server, follow the actions that you need to change the role, and then bring up the server.

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

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 Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached to the Root Server with Workstations will be scanned for. 2. Ensure that you enable the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations attached to will be done.
Intermediate Server	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 this server. 2. If a Database Location policy is associated with a Root Server, remove the policy.
Intermediate Server with Database	Perform the following task: 1. After changing the role, configure the Roll-Up to specify the next-destination server for roll-up of data from this server.
Intermediate Server with Database & Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for. 2. Ensure that you enable the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations will be done. 3. Configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server.

To change the role of the Root Server to ...	Tasks:
Intermediate Server with Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for. 2. Ensure that you enable the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations will be done. 3. Configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server. 4. If a Database Location policy is associated with the Root Server, remove the policy.
Leaf Server, Leaf Server with Database, or Standalone Server	Workstation Inventory does not allow you to change the Root 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 Workstation Inventory component.

Changing the Role of the Root Server with Workstations

Follow the actions specified in the following table:

To Change the Role of the Root Server with Workstations to ...	Tasks:
Root Server	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Root Server with Workstations or reconfigure the Inventory policy.
Intermediate Server	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 this server. 2. If the Database Location policy is associated with the Root Server with Workstations, remove the policy.
Intermediate Server with Database	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 this server. 2. Before changing the role, if the Workstation Inventory policy is associated with the Root Server with Workstations, remove the policy for those workstations attached to this server or to the lower-level servers that roll up to this server.
Intermediate Server with Database & Workstations	Perform the following tasks: 1. Ensure that you enable the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations will be done. 2. Configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server.

To Change the Role of the Root Server with Workstations to ...	Tasks:
Intermediate Server with Workstations	Perform the following task: 1. Before changing the role, if the Database Location policy is associated with the Root Server with Workstations, remove the policy.
Leaf Server, Leaf Server with Database, or Standalone server	Workstation Inventory does not allow you to change the Root 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 Workstation 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 to this role, remove the Roll-Up policy. 2. Configure the Database Location policy.
Root Server with Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy for those workstations attached to this server. 2. Before changing to this role, remove the Roll-Up policy. 3. Configure the Database Location policy.
Intermediate Server with Database	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Database & Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that all the workstations associated to this Inventory Service object, and also those workstations associated to the lower-level servers that roll up to this server will be scanned for. 2. Configure the Database Location policy.
Intermediate Server with Workstations	Perform the following task: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for.
Leaf Server, Leaf Server with Database, or Standalone server	Workstation 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 Workstation 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 to this role, remove the Roll-Up policy.
Root Server with Workstations	Perform the following tasks: 1. Before changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for. 2. Remove the Roll-up policy.
Intermediate Server	Perform the following task: 1. Before changing the role, if the Database Location policy is associated with the Intermediate Server with Database, remove the policy.
Intermediate Server with Database & Workstations	Perform the following task: 1. After changing the role, configure the Workstation Inventory policy so that the workstations attached will be scanned for.
Intermediate Server with Workstations	Perform the following tasks: 1. Before changing the role, if the Database Location policy is associated with the Intermediate Server with Database, remove the policy. 2. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for.
Leaf Server, Leaf Server with Database, or Standalone server	Workstation 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 Workstation Inventory component.

Changing the Role of the Intermediate Server with Database & Workstations

Follow the actions specified in the following table:

To Change the Role of the Intermediate Server with Database & Workstations to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. Remove the Workstation Inventory policy associated with the server so that the workstations will not send the scan files to this server.

To Change the Role of the Intermediate Server with Database & Workstations to ...	Tasks:
Root Server with Workstations	Perform the following task: 1. Before changing to this role, remove the Roll-Up policy.
Intermediate Server	Perform the following tasks: 1. Remove the Workstation Inventory policy associated with the lower-level servers that roll-up to the Intermediate Server with Database & Workstations server. 2. Remove the Database Location policy.
Intermediate Server with Database	Perform the following task: 1. Remove the Workstation Inventory policy of the Intermediate Server with Database & Workstations or reconfigure the policy.
Intermediate Server with Workstations	Perform the following task: 1. Before changing the role, remove the Database Location policy associated with the Intermediate Server with Database & Workstations.
Leaf Server, Leaf Server with Database, Standalone Server, or Intermediate Server with Workstations	Workstation 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 Workstation Inventory component.

Changing the Role of the Intermediate Server with Workstations

Follow the actions specified in the following table:

To Change the Role of the Intermediate Server with Workstations to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. Remove the Workstation Inventory policy associated with the server so that the workstations attached will not send the scan files to this server. 3. After changing the role, configure the Database Location policy for the server.
Root Server with Workstations	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. After changing the role, configure the Workstation Inventory policy for those workstations attached to the lower-level server that roll up to this server. 3. Configure the Database Location policy

To Change the Role of the Intermediate Server with Workstations to ...	Tasks:
Intermediate Server	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy associated with the server.
Intermediate Server with Database	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy associated to the server attached to this Inventory Service object. 2. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Database & Workstations	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Leaf Server, Leaf Server with Database, Standalone Server, or Intermediate Server with Workstations	Workstation 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 Workstation Inventory component.

Change 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 to this role, remove the Roll-Up policy. 2. Remove the Workstation Inventory policy associated with the server. 3. After changing the role, configure the Database Location policy for the Root Server.
Root Server with Workstations	Perform the following tasks: 1. After changing the role, Configure the Database Location policy. 2. Remove the Roll-Up policy.
Intermediate Server	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy for those workstations associated with the server or reconfigure.
Intermediate Server with Database	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy for those workstations associated with the lower-level servers that roll up to this server or reconfigure the policy. 2. After changing the role, configure the Database Location policy for the server.

To Change the Role of the Leaf Server to ...	Tasks:
Intermediate Server with Database & Workstations	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Workstations	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 the server.
Standalone Server	Perform the following task: 1. Remove the Roll-up policy.

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: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Leaf Server with Database. 2. Remove the Roll-up policy.
Root Server with Workstations	Perform the following task: 1. Remove the Roll-up policy.
Intermediate Server	Perform the following task: 1. Before changing to this role, remove the Workstation 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 to this role, remove the Workstation Inventory policy associated with the Leaf Server with Database.
Intermediate Server with Database & Workstations	This change of role does not require any specific policy modifications.
Intermediate Server with Workstations	Perform the following task: 1. Before changing to this role, remove the Database Location policy associated with the Leaf Server with Database.
Leaf Server	Perform the following task: 1. Before changing to this role, remove the Database Location policy associated with the Leaf Server with Database.

To Change the Role of the Leaf Server with Database to ...	Tasks:
Standalone Server	Perform the following task: 1. Before changing to this role, remove the Roll-Up policy.

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 to this role, remove the Workstation Inventory policy associated with the Standalone Server.
Root Server with Workstations	This change of role does not require any specific policy modifications.
Intermediate Server	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy and the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy.
Intermediate Server with Database	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory 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 Database.
Intermediate Server with Database & Workstations	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 & Workstations.
Intermediate Server with Workstations	Perform the following tasks: 1. Before changing to this 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 Workstations.
Leaf Server	Perform the following tasks: 1. Before changing to this 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.

To Change the Role of the Standalone Server to ...	Tasks:
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.

Assigning Trustees to the Database Object

You must have administrative privileges for any inventory operations from ConsoleOne. An administrator can assign the users as trustees of the database object.

To assign trustees to the database object:

1. Log in as an administrator to the tree and context where you have installed ZfD.
2. In ConsoleOne, right-click the existing database object > click Trustees of this object > click Add Trustee.
3. Browse to the user objects or container that contains the user objects > click the user objects that need to be the trustees or click the container > click OK.
4. Click All Attributes Rights > select Read rights > click OK twice.

Migrating ZENworks 2 Inventory Information

In ZfD 3.2, you can use the existing inventory information of ZENworks 2.

If you have configured ZENworks 2 for a large number of workstations, you need not disrupt the inventory process when upgrading to ZfD 3.2. You need to follow this procedure to continue using the inventory information:

1. Migrate the Inventory policies in ZfD 3.2.
2. Migrate the ZENworks 2 database

The following sections will help you migrate your ZENworks 2 inventory information to ZfD 3.2.

- "Using ZENworks 2 Inventory Policies" on page 391
- "Migrating the Inventory Information from the ZENworks 2 Database" on page 392
- "Scenarios: Migration of ZENworks 2 Inventory Information" on page 395

After successfully migrating the policies and the database, the inventory information of ZENworks 2 should exist in ZfD 3.2. Also, the inventory information in ZENworks 2 and ZfD 3.2 can coexist. The workstation inventory is stored in the ZENworks 2 database as specified in the ZENworks 2 Inventory policies. The inventory information is also stored in the ZfD 3.2 database. You can configure the Inventory policy settings to ensure a full scan of the workstations.

Using ZENworks 2 Inventory Policies

If you have a large number of workstations associated with Inventory policies in ZENworks 2, you can continue to use the same policies in ZfD 3.2. Migrating ZENworks 2 Inventory policies is an easy method of importing the same policy settings for the associated workstations to ZfD 3.2.

The Inventory policy migration tool associates the workstations that were associated with the Inventory policy in ZENworks 2. After you migrate, you may need to further configure the policy settings in ZfD 3.2.

The Inventory policy migration tool does not remove the Inventory policies in ZENworks 2.

To migrate ZENworks 2 Inventory policies:

1. Migrate the Workstation Inventory Package in ZfD 3.2. See Migrating Workstation Inventory from ZENworks 2 in Workstation Inventory in *Deployment*.
2. In ConsoleOne, click the Inventory Service object to which the workstations are attached.

   You must select an Inventory Service object that supports the role of an inventory server.
3. Click Tools > Inventory Policy Migration.
4. Specify the following options:

   Server Address IP/DNS: If your ZENworks 2 inventory server is a NetWare® 4.*x* server, specify the Server Address.

   NDS Search Context: Specify the context for searching the Workstation Inventory object. By default, this tool will search the Workstation object in the current root context.
5. Click Find.

   If any ZENworks 2 Inventory policies are found, these policies are listed in the Reports window.
6. Click Migrate.

   All the listed Inventory policies will be migrated. You can see the list of the Inventory policies that were migrated in the Report window.

   To ensure that the migration is successful, open the Inventory policy in ZfD.

   In ConsoleOne, double-click the Inventory Service object for which you have migrated the policies > click the Workstation Inventory policy tab. You will see the same inventory settings as specified in ZENworks 2.

Migrating the Inventory Information from the ZENworks 2 Database

The Database Migration tool migrates the existing inventory information from the ZENworks 2 database to the ZfD 3.2 database. The Migration tool provides an easy way to migrate a large amount of existing data into the ZfD 3.2 database. Once you migrate the database, you can choose when you want to rescan the workstations in the inventory policies.

The migration tool migrates the following list of inventory information:

- Mouse, BIOS, bus, user, inventory scanner, keyboard, unitary computer system, monitor, software, processor, CD ROM, operating system, parallel port, serial port, general information, hard disk, floppy disk, NIC information, IP address and subnet mask, IPX™ information, DNS information, modem information, and memory information.

NOTE: The migration tool does not migrate the DMI scan data.

When you install ZfD 3.2, the migration tool that uses the DBMIGRATE.NCF file is installed on NetWare servers. This file is in the SYS:\SYSTEM directory on a NetWare server and in the PUBLIC\ZENWORKS\WMINV\BIN on a Windows* NT/2000 server. On Windows NT/2000, the batch file, DBMIGRATE.BAT is installed.

The contents of DBMIGRATE.NCF file are as follows:

```
envset
JDBC_DRIVER=SYS:\PUBLIC\ZENWORKS\LIB\JDBCDRV.ZIP;SYS:\PUBLIC\ZENWORKS\LIB\C
LASSES111.ZIP

envset
WORKING_PATH=SYS:\PUBLIC\ZENWORKS\WMINV\LIB\STATUSLOG.JAR;SYS:\PUBLIC\ZENWO
RKS\WMINV\LIB\DESKTOPCOMMONUTILITY.JAR; SYS:\PUBLIC\ZENWORKS\MWUTILITY.JAR

envset CLASSPATH=.;$JDBC_DRIVER;$WORKING_PATH;$CLASSPATH

#

# Running the DBMigrate Utility

# *********

# To run the DBMIGRATE Utility, do the following:

#

#1. Check if the environment variable JDBC_DRIVER points to the path where
JDBCDRV.ZIP is present.

#2. Check if the environment variable WORKING_PATH points to the path where
ZENINVSERVER.JAR, STATUSLOG.JAR, and DESKTOPCOMMONUTILITY.JAR are present.

# 3. Enter the IP address ZENworks for Desktops 3 Inventory database server
after the switch -DBLOC. For example, $CLASSPTH -dbloc 164.99.156.184

# 4. Enter the IP address of ZENworks 2 Inventory database after the switch
-zen2dbloc. For example, $classpath -dbloc 164.99.156.134 -zen2dbloc
164.99.156.135

# **********

# If the ZENworks for Desktops 3 Inventory database is running on Oracle, then
uncomment the below the line and comment the Sybase line.

# java -ns com.novell.zenworks.desktop.inventory.migration.database.Loader -
classpath $classpath;$tmppath -nds -dbloc 164.99.135.198 -zen2dbloc
164.99.156.42 -oracle -sid abuorcl -Logfilename sys:\etc\dbmigrte.log

# If the ZENworks for Desktops 3 Inventory database is running on Sybase, then
uncomment the below line and comment the above Oracle line.

-classpath $classpath;$tmppath -nds -dbloc 164.99.156.42 -zen2dbloc
164.99.145.53 -Logfilename SYS:\ETC\DBMIGRATE.LOG
```

The contents of DBMIGRATE.BAT are as follows:

```
echo .

IF "%1"=="" goto no3

IF "%2"=="" goto no3

call InvEnv.bat

java -
Xbootclasspath:..\..\lib\vbjapp.jar;..\..\lib\vbjorb.jar;%java_dir%\lib\rt.
jar -mx128m -classpath %tmppath%;%classpath%
com.novell.zenworks.desktop.inventory.migration.database.Loader -nds -dbloc
%1 -zen2dbloc %2 -logfilename dbmigrate.log

goto end
```

```
:no3

echo Migration source and destination addresses not specified.

echo Incorrect parameters.

echo USAGE : DBMIGRATE "ZfD3 database IPAddress" "ZfD2 database IPAddress"

:end

echo .
```

You must modify the file to specify the location of the ZENworks 2 database server and the ZfD 3.2 database and the path settings.

To migrate a database from ZENworks 2:

1. Ensure that the Inventory database is installed when you install ZfD 3.2.
2. Stop the Inventory Gatherer in ZENworks 2.

   Also, ensure that all the .STR files are stored in the database. If the .STR file data is updated in the database, the directory containing the .STR files (ZENWORKS\STRDIR) will be empty. For more information about unloading the Inventory Gatherer, see the ZENworks 2 documentation Web site (http://www.novell.com/documentation).
3. In ZfD 3.2, stop the Service Manager.
4. Modify the settings on the server as follows:

   - If the database server is on a NetWare server, modify the settings in the DBMIGRATE.NCF file. At the database server console, enter **DBMIGRATE.NCF**. On a NetWare 4 server console, enter **DBMIGRAT.NCF**.
   - If the database server is on a Windows NT/2000 server, at the database server console, run the DBMIGRATE batch file for Sybase database:

     dbmigrate *ipaddress_of_zfd3.2_database server* *ipaddress_of_zenworks2_database_server*

     By default, the commands in the DBMIGRATE.BAT file apply to the Sybase* database on Windows NT/2000. If you have an Oracle* database you must modify the following entry in DBMIGRATE.BAT:

     ```
     java -Xbootclasspath:..\..\lib\vbjapp.jar;
     ..\..\lib\vbjorb.jar;%java_dir%\lib\rt.jar -mx128m -classpath
     %tmppath%;%classpath% com.novell.zenworks.
     desktop.inventory.migration.database.Loader -nds -dbloc %1 -zen2dbloc
     %2 -logfilename dbmigrate.log
     ```

     Between these two parameters, %2 and -logfilename, type **-oracle -sid *sid***.

     sid refers to the Oracle database ID. An example for sid is orcl.

     Save the file. On the database server console, run the DBMIGRATE batch file:

     dbmigrate *ipaddress_of_zfd3.2_database server* *ipaddress_of_zenworks2_database_server*

You will see a list of workstations that have been successfully migrated into the ZfD database.

On NetWare server, the database migration tool creates a log file (DBMIGRATE.LOG) in the SYS:\ETC directory. On Windows NT/2000, this log file is in the same directory where you run the DBMIGRATE.BAT file.

If you have installed more than one ZfD 3.2 database, you must migrate the ZENworks 2 database to each database, repeating the database migration procedure for each database.

The database information from the ZENworks 2 database will not be removed.

Deleting the Inventory Information from ZENworks 2 database

You can remove the inventory information from the ZENworks 2 Inventory database, ZENINV.DB, using the Delete tool.

Before you run this tool, ensure that the ZENworks 2 and ZfD 3.2 databases are not loaded. You can run this tool on any inventory server that has the ZENworks 2 and ZfD 3.2 databases connected to the server.

To run this tool:

1. Modify the MGMTDBS.NCF file.

   1a. Insert the -d parameter in the MGMTDBS.NCF after the dvsrv7 switch.

   The contents of this file should be as follows:

   ```
   dbsrv7 -d -gm 50
   ```

   1b. Save the changes.

2. Load the ZfD 3.2 database. At the server prompt, enter **mgmtdbs**.

   This will also load the ZENworks 2 database (ZENINV.DB).

3. Stop the Naming server.

   On NetWare servers, at the inventory server prompt, to view the ID of the Naming server, enter **JAVA -show**

   Enter **JAVA -KILL*id_of_the_naming_server***

4. At the server prompt, enter **zen2remove**.

   Displays the message that the inventory information is being deleted.

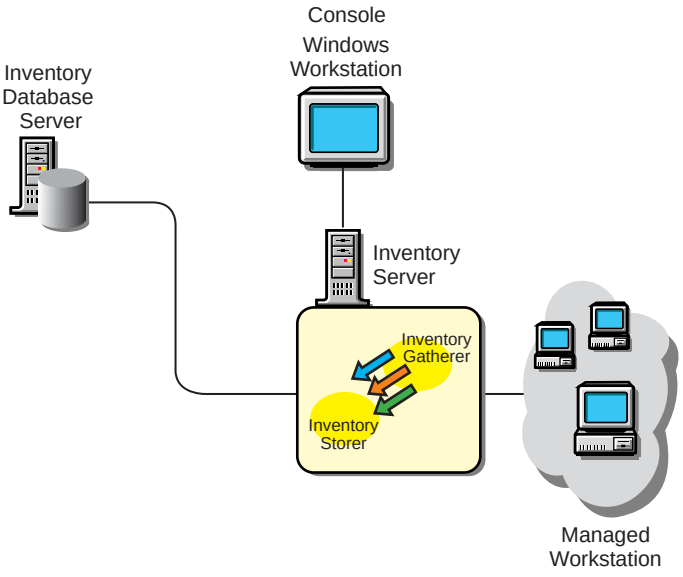
   On completing the removal of inventory information from the ZENINV.DB database, the status of the deletion is displayed. To view the log file for the Delete tool, open the file DROPZEN2.LOG file in the SYS:\ETC directory.

5. Remove the modifications in the MGMTDBS.NCF file.

   Delete the -d parameter in the MGMTDBS.NCF file and save the changes.

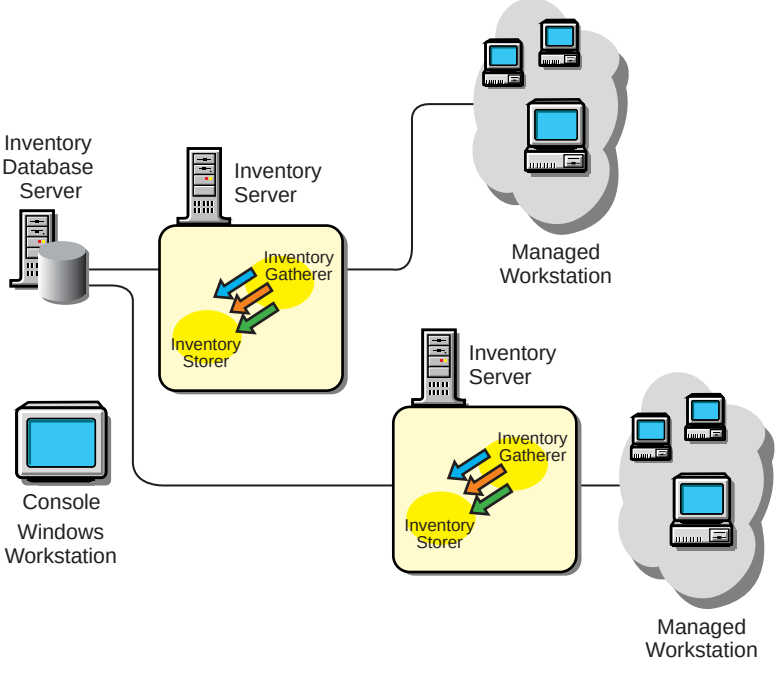
Scenarios: Migration of ZENworks 2 Inventory Information

In this type of inventory configuration in ZENworks 2, the inventory server components and the database are located on different servers. The scenario is illustrated in the following figure:



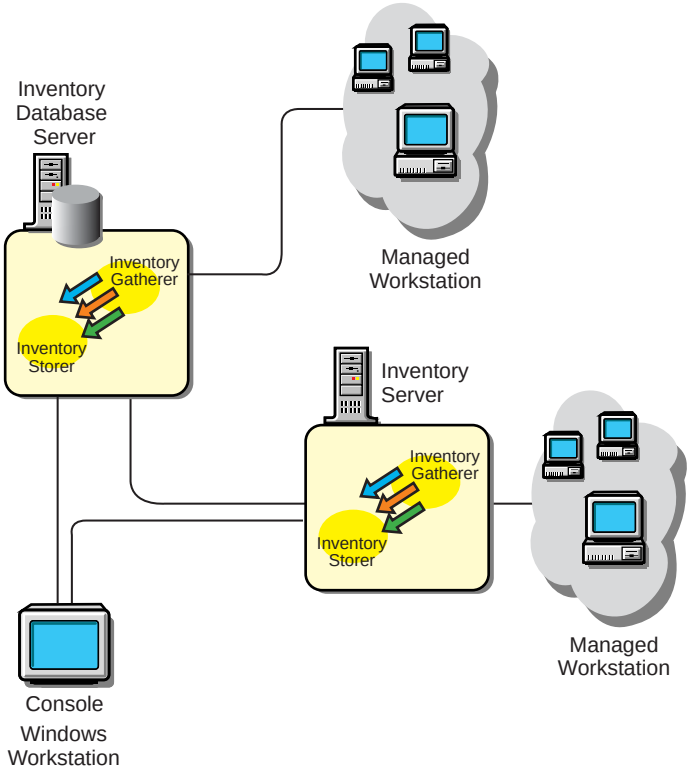
To migrate the inventory information, you need to migrate the Workstation Package and the Workstation Inventory policies associated with the inventory server. Later you migrate the Inventory database and configure the Database Location policy.

In another type of inventory configuration in ZENworks 2, there is more than one inventory server. These inventory servers are connected to one Inventory database server. The scenario is illustrated in the following figure:



To migrate the inventory information, you migrate the Workstation Package and the Workstation inventory policies associated with each inventory server. Later you migrate the Inventory database and configure the Database Location policy.

In the following inventory configuration in ZENworks 2, there is more than one inventory server. Also, an inventory server has an Inventory database; the other inventory servers connect to this inventory server with the database. The scenario is illustrated in the following figure:



To migrate the inventory information, you need to migrate the Workstation Package and the inventory policies associated with the inventory server with a database. Later, you migrate the other inventory servers. You must stop the Inventory components (Inventory Gather and Inventory Storer) on the server.

Setting up the Inventory Database

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

- “Configuring the Sybase ODBC Driver for ZfD 3.2 Inventory Database” on page 409
- “Optimizing the Performance of the Oracle Database” on page 398
- “Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool)” on page 398
- “Using an Optimal Database Cache Size on the Inventory Database Server to Improve Performance” on page 400
- “Using the Inventory Database Service Configuration Tool for Windows NT/2000” on page 400
- “Deleting the Inventory Information from the Inventory Database” on page 401
- “Synchronizing the Inventory Database with eDirectory (NDS-DB Sync Tool)” on page 401
- “Backing Up the Inventory Database” on page 402

- "Using the ZfD 3.2 Inventory Database on Oracle 8i for Linux and Solaris" on page 405
- "Upgrading the ZfD 3.2 Inventory Database on Oracle 8.1.x" on page 408

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 ZfD 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.

Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool)

If there are more volumes on the multiple physical disks of the database server, placing the Sybase database space files on separate volumes improves the performance while accessing the database.

When you install the Sybase database component of ZfD 3.2, the System Database file and the Database Spaces are installed in the volume 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 and MGMTDB9.DB) contain the inventory information.

The ALTERDB.PROPS file is installed on the database server in the PUBLIC\ZENWORKS\WMINV\PROPERTIES. You modify the sections in the file to specify the location of the Database Spaces on the volumes.

The contents of the ALTERDB.PROPS file is as follows:

```
#Database Space Properties

count=9

mgmtdb1=volume_location_of_mgmtdb1

mgmtdb2=volume_location_of_mgmtdb2

mgmtdb3=volume_location_of_mgmtdb3

mgmtdb4=volume_location_of_mgmtdb4
```

mgmtdb5=*volume_location_of_mgmtdb5*

mgmtdb6=*volume_location_of_mgmtdb6*

mgmtdb7=*volume_location_of_mgmtdb7*

mgmtdb1=*volume_location_of_mgmtdb8*

mgmtdb1=*volume_location_of_mgmtdb9*

.....

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 server.
3. Manually move the Database Space files on the server volume.

   Move the Database Space files in the following way for better performance:

   - MGMTDB1 and MGMTDB2 on the same volume
   - MGMTDB3 and MGMTDB6 on the same volume
   - MGMTDB5 and MGMTDB7 on the same volume
   - MGMTDB8 and MGMTDB4 on the same volume
   - MGMTDB9 on a volume
4. Modify the location of the nine Database Space files in the ALTERDB.PROPS file.

   For example, enter **mgmtdb3=*SYS:**ZENWORKS**DATABASE***
5. Load the database. Enter **mgmtdbs** on NetWare servers. On Windows NT/2000 servers, run the database service.

   5a Stop the Naming server.

   On NetWare servers, at the inventory server prompt, to view the ID of the Naming server, enter **JAVA -show**.

   Enter **JAVA -KILL*id_of_the_naming_server***.

   On Windows NT/2000, close the Naming server window.

   Ignore the error messages displayed on the console. These messages are displayed because the database space files are not loaded.
6. Ensure that the Database Location policy has been configured.
7. On the server console, run the AlterDBSpace service. Enter **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 while loading the database there are no errors. Errors indicate that the specified volume location of the database space files are incorrect or do not exist. Ensure that the path of the Database Spaces is correct in the ALTERDB.PROPS file and repeat the procedure to organize the Database Spaces.

   IMPORTANT: If you place the Database Spaces in different volumes, the log file should be placed in the same volume as the System database file (MGMTDB.DB).

Using an Optimal Database Cache Size on the Inventory Database Server to Improve Performance

Improving 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 servers. The default database cache size is 16 MB; however, this database cache size may not be adequate for large databases with more than 10,000 workstations.

You should change the database cache size to an optimum size. We recommend a database cache size that is one-fourth of the database 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 a NetWare server:

1. Close all connections to the Inventory database.
2. Quit the Sybase server.
3. Open the MGMTDB.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 server:

1. Run the NTDBCONFIGURE.EXE file from PUBLIC\ZENWORKS.
2. Modify the -c parameter.
3. Save the file.
4. Restart the server so that the Inventory database service (Adaptive Service Anywhere - ZENworks for Desktops 3) starts up.

Using the Inventory Database Service Configuration Tool for Windows NT/2000

The Inventory Database service configuration tool lets you modify the startup parameters to optimize the Sybase database on Windows NT/2000 servers.

To run this tool:

1. Run NTDBCONFIG.EXE located in the ZENWORKS directory on the database server.
2. Modify the parameters as per the Sybase specification.

   Ensure that you specify valid values for the parameters.
3. Click OK.
4. Restart the database service (ASA Adaptive Anywhere).

Sybase Database Startup Parameters

The parameters 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 name of the database server. For example, -n WARTHOG names the server WARTHOG.
- **-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.

Deleting the Inventory Information from the Inventory Database

If you delete the Workstation object in ConsoleOne, the inventory information for that Workstation object is deleted from the Inventory database server it is attached to. Also, the inventory information is deleted from all the next-level servers for that particular Inventory database server.

Synchronizing the Inventory Database with eDirectory (NDS-DB Sync Tool)

ZfD provides a database sync tool to synchronize the Workstation objects stored in the Inventory database with the Workstation objects in eDirectory. This tool removes the workstations that do not exist in eDirectory from the Inventory database. The excess workstations in the Inventory database exist because these workstations may have been deleted from eDirectory; however, the corresponding workstations were not removed from the database.

Use this tool regularly to maintain the database in a consistent state with eDirectory. You must run this tool for each Inventory database.

HINT: You must ensure that the Service Manager is loaded when you run the Inventory database sync tool.

To run this tool:

1. Generate a lookup file in the NDS Lookup Phase.

   ZfD compares the list of workstations in the database with those in eDirectory and generates a lookup file. This file contains the list of workstations to remove from the database.

   The server property file contains the [NDSLookupForDB Service] section with the lookup filename in the ARGUMENTS parameter. The default lookup filename (WSDELETE.LOK) file is located in the PUBLIC\ZENWORKS\WMINV\LOG directory.

   The lookup filename in this section and the [DBDelete Service] section of the server property file should be the same.

   1a. At the inventory server console prompt, enter:

```
StartSer NDSLookupForDB
```

2 Delete workstations in the lookup file from the Inventory database in the Database Delete Phase.

2a In the [DBDelete Service] section of the properties file, specify the lookup filename.

2b At the inventory server console, enter:

```
StartSer DBDelete
```

The WSDELETE.LOG log file located in the \PUBLIC\ZENWORKS\WMINV\LOGS directory contains the status of deletion. This file contains information as to whether the database was synchronized successfully with eDirectory.

Backing Up the Inventory Database

ZfD provides an option to back up the Inventory database from the server. We recommend that you back up the database on a weekly basis. However, if you are tracking the inventory of workstations frequently, increase the frequency of backup.

The following sections contain additional information to help you back up the Inventory database.

- "Backing Up the Inventory Database (Sybase)" on page 402
- "Backing Up the Inventory Database (Oracle)" on page 403

Backing Up the Inventory Database (Sybase)

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

To use this tool:

- Run the backup tool on the inventory server to which you have associated a database server. If you deployed more than one database server, you must run the backup tool for each database server.
- The database that you back up should be configured in the Database Location policy.
- In the [DBBackup Service] section of the server property file, you modify the location of the backup destination in the ARGUMENTS parameter. The backup files are relative to the SCANDIR path. For example, if the SCANDIR path is SYS:\ZENWORKS\SCANDIR, the database is backed up in SYS:\ZENWORKS\DATABASE*directory_you_specify*. You must the modify the server property file located on the server on which you are running the backup tool. Modify the server property file corresponding to the role of the server. For example, if you are running the backup tool on the Leaf Server with Database, modify the server property file, LEAF_DB_WKS.PROPERTIES.
- Ensure that the Service Manager is loaded when you run the backup tool.

To run the backup tool on NetWare or Windows NT/2000 servers,

1 At the inventory server console, enter **`StartSer DBBACKUP`**.

View the status of the backup in the backup log file.

The database will be copied to SYS:\ZENWORKS\DATABASE*directory_you_specify*.

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/2000: Stop the Sybase service (Adaptive Service Anywhere - ZENworks for Desktops 3).
3. Copy the backup files, overwriting the working database files.
4. Restart the database server.

The backup tool creates a log file, BACSTATUS.TXT, located in the ZENWORKS\DATABASE 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.

Backing Up the Inventory Database (Oracle)

ZfD provides an option to back up the Inventory database from the server. However, if you have a database backup and restore method such as Recovery Manager, we recommend that you continue to use your existing backup method.

To back up 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 NetWare server with Oracle 8i, enter **svrmgr31**.

   On NetWare server with Oracle 8.0.4, enter **svrmgr30**.

   On Windows NT/2000 server with Oracle 8i Enterprise Edition, from the taskbar, click Start > Run > enter **svrmgrl**.
3. Enter the following commands:

   set instance *databasevername-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 in Workstation Inventory in *Deployment*.

   For example, **set instance *austr-zfd3-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 in Workstation Inventory in *Deployment*.

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

   This displays the list of the datafiles that Workstation 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 ZFD3\ORACLE 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 8i, enter **load DBV81.NLM FILE=***path_to_the_database_file* **BLOCKSIZE=*4096***

To verify the database integrity on a NetWare server with Oracle 8.0.4, enter **load DBV80.NLM FILE=*path_to_the_database_file* BLOCKSIZE=*4096***.

To verify the database integrity on a Windows NT/2000 server with Oracle 8i, enter **DBV.EXE FILE=*path_to_the_database_file* BLOCKSIZE=*4096***

Example: enter **DBV.EXE FILE=*d:\zfd3\oracle\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, 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 8i, enter **svrmgr31**.

   On a NetWare server with Oracle 8.0.4, enter **svrmgr30**.

   On a Windows NT/2000 server with Oracle 8i 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 \ZFD3\ORACLE\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=oracle:\zfd3\oracle\database\INIT.ORA
  logfile group 1 'oracle:\zfd3\oracle\database\log1.ora' size 256K,
  logfile group 2 'oracle:\zfd3\oracle\database\log2.ora' size 256K
  datafile 'oracle:\zfd3\oracle\database\sys1.ora',
  'oracle:\zfd3\oracle\database\rbs1.ora',
  'oracle:\zfd3\oracle\database\cim1.ora',
  'oracle:\zfd3\oracle\database\cim2.ora',
  'oracle:\zfd3\oracle\database\cim3.ora',
  'oracle:\zfd3\oracle\database\cim4.ora',
  'oracle:\zfd3\oracle\database\cim5.ora',
  'oracle:\zfd3\oracle\database\cim6.ora',
  'oracle:\zfd3\oracle\database\cim7.ora',
  'oracle:\zfd3\oracle\database\cim8.ora',
  'oracle:\zfd3\oracle\database\cim9.ora',
  'oracle:\zfd3\oracle\database\tmp1.ora'
  ```

  Save the changes.

7. Load the restored database.

Using the ZfD 3.2 Inventory Database on Oracle 8i for Linux and Solaris

Ensure that the following requirements are met:

- Oracle version

  On Linux* 6.0 or above: Oracle 8i (8.1.5 or above) Enterprise Edition

  On Solaris* 6.2 or above on Sparc*/Intel*: Oracle 8i (8.1.5 or above) Enterprise Edition

- System requirements

  Hard disk free space: 500 MB or above

  Primary memory: 256 MB or above

Follow the instructions in these sections:

- "Starting the Inventory Database" on page 406
- "Initializing the Inventory Database" on page 407

- “Configuring the Inventory Database” on page 407

Starting the Inventory Database

Follow these steps:

1. Log in to the Linux box as Oracle DBA user.
2. Create a ZfD3 directory. Change to this directory.
3. Extract the file from the \ZENWORKS\PRODUCTS\DATABASE \ORACLE directory into ZfD3 directory.

   On a Linux / Solaris(Intel) server, extract INVORACLE8IUNIXINTEL.TAR.GZ file.

   On a Sparc Solaris server, extract INVORACLE8ISPARCSOL.TAR.GZ file.
4. Enter the following commands:

   On a Linux/Solaris(Intel) server, enter **`$ gunzip InvOracle8iUNIXintel.tar.gz`**

   Enter **`$ tar -xvf InvOracle8iUNIXintel.tar`**

   On a Sparc Solaris server server, enter **`$ gunzip InvOracle8iSparcSol.tar.gz`**

   Enter **`$ tar -xvf InvOracle8iSparcSol.tar`**

   This extracts the Inventory database in the *home*/ZfD3/zenworks directory, where *home* is the home directory of the Oracle user you logged in as.
5. Edit the _START.SQL file in ZfD3/ZENWORKS directory. This file contains commands to start the Inventory database.

   If required, edit the file based on your Oracle settings. For example, if you are starting the Inventory database along with other Oracle databases, create a Oracle database instance for the Inventory database. Set the Oracle instance name in _START.SQL. Otherwise, shut down any existing databases.
6. Ensure that Oracle is up and running.

   To run the Oracle Server Manager, enter **`$ svrmgr1`**

   This command loads the Oracle Server Manager.
7. From the Server Manager, start the Inventory database.

   At the Server Manager prompt, enter the following commands:

   Type **`@$HOME/ZfD3/zenworks/_start`**

   This command starts the Oracle instance, mounts the Inventory database, and displays the following output information.

   ```
   ORACLE instance started.Total System Global Area ... Fixed Size
   ... Variable Size... Database Buffers... Redo Buffers... Database
   mounted. Database opened.
   ```

   While mounting the database, you may see the following error: `End-of-file on communication channel`. Resolve this error by creating a new control file and using it to open the Inventory database.

   To create a new control file:

   7a Open the ALTERCTRL.SQL file in the ZENWORKS directory.

   7b Ensure that the specified path settings in the file are correct.

7c Run the Oracle Server Manager and connect as an Internal user.

7d At the Server Manager prompt, run ALTERCTRLSQL.

Type **`@$HOME/ZfD3/zenworks/alterctrl`**

7e Start the Inventory database.

Initializing the Inventory Database

Before using the Inventory database, initialize the database. Initializing the Inventory database assigns a unique site ID and site name to the database. Using these details, the Inventory database can be identified at the enterprise level.

Follow these steps:

1. Ensure that the Inventory database is mounted.
2. Specify a site ID and site name that uniquely identify your Inventory database in the ZENworks tree.

   Open the _DBINIT.SQL file in the ZFD3/ZENWORKS directory. This file contains a dummy site ID and site name as:

   ```
   siteid:=255;
   siteName:='example-site';
   ```

   Change the site ID and site name variables with your site ID and site name and save the file.

   IMPORTANT: Do not use an already existing site ID and site name.
3. Run the Oracle Server Manager. At the Server Manager prompt, enter the following commands:

   3a Enter **`connect mw_dba/novell`**

   3b Enter **`@$HOME/ZfD3/zenworks/_dbinit`**

   This command initializes the database with the provided site ID and site name. To verify this, at the Server Manager prompt, enter **`select * from zenworks.site;`**

   While initializing the database, you may see the following error: `Declare * ORA-06553: PlS-908: The stored format of SYS.STANDARD is not supported by this release.`

   This error indicates that the Oracle version that is in use is later than 8.1.5.

   To correct this problem, upgrade the Inventory database to initialize the Inventory database.

   Follow the steps in "Upgrading the ZfD 3.2 Inventory Database on Oracle 8.1.x" on page 408.

Configuring the Inventory Database

Create an identity for the Oracle Inventory database server in the ZENWORKS tree. This makes the database available for access by the inventory components.

1. Create a ZENworks Inventory Database object and a Database Location policy. In the policy, indicate the Oracle Inventory database server as the designated server for hosting Inventory database. See Configure the Policies for the Database in Workstation Inventory in *Deployment*. Ensure that you specify the IP address of the Oracle Inventory database server in the Database policy.

Upgrading the ZfD 3.2 Inventory Database on Oracle 8.1.*x*

When you are mounting ZfD 3.2 Inventory database on a server with Oracle 8.1.*x*, initializing the database using _dbinit.sql may be unsuccessful and you may see the following error message:

```
DECLARE * ORA-06553: PLS-908: The stored format of SYS.STANDARD is not
supported by this release.
```

In this scenario, it is not possible to use the Inventory database. To resolve this problem, upgrade the Inventory database to the existing Oracle version.

1. Configure and mount the Inventory database.

   On NetWare and Windows NT/2000 servers, follow the instructions in Configuring the Inventory Database for Oracle in Workstation Inventory in *Deployment*. Do not follow the last step in the procedure, which instructs you initialize the database on the server.

   On UNIX* servers, see "Using the ZfD 3.2 Inventory Database on Oracle 8i for Linux and Solaris" on page 405.

2. Run the Oracle Server Manager.

   On a NetWare server: From the server prompt, enter **`load svrmgr31.nlm`**.

   On a Windows NT/2000 server: From the taskbar, click Start > Run. Enter **`svrmgrl.exe`**.

   On a UNIX server: Log in as an Oracle administrator. At the login prompt, run **`svrmgrl`**.

3. At the Oracle Server Manager prompt:

   Enter **`connect internal`**/*password* where *password* is the Oracle internal user password. On NetWare or Windows NT/2000 servers, enter **`spool`** ***`path`*****`\upgrade81X.log`**

   `@%`*`oracle_home`*`%\rdbms\admin\u801050.sql`

   On UNIX-based servers, enter the following commands: **`spool $HOME/upgrade81X.log`**
   `@%`*`oracle_home`*`%\rdbms\admin\u801050.sql`

   The upgrade tool starts upgrading the Inventory database. Wait until the upgrade tool completes the upgrade. The upgrade tool may take some time to complete.

4. At the Oracle Server Manager prompt,

   4a Enter **`spool off`**

   4b Enter **`disconnect`**

   4c Enter **`exit`**

   The Inventory database will be upgraded to Oracle 8.1.*x*. The status of update is logged in the UPGRADE81X.LOGfile.

5. Initialize the Inventory database.

6. Update with the Oracle JDBC driver.

   The Oracle JDBC driver that is shipped as part of ZfD 3.2 is for Oracle 8i (8.1.5). If you are using a later version of Oracle, we recommend that you use the JDBC driver for the same version. Oracle JDBC driver is located on the server in the *oracle_home*/JDBC/LIB directory, where *oracle_home* is the directory where Oracle is installed. The name of the driver file is CLASSES111.ZIP. Alternatively, you can download the file from Oracle Web site. After you get the correct JDBC driver, update the file on ZfD 3.2 inventory servers and consoles.

Follow these steps:

1. To update all ZfD 3.2 consoles, copy CLASSES111.ZIP to the \CONSOLEONE\1.2\CONSOLEONEEXT and CONSOLEONE\1.2\LIB\ZEN directories
2. To update all inventory servers, copy CLASSES111.ZIP to the ZENWORKS\LIB directory.

Configuring the Sybase ODBC Driver for ZfD 3.2 Inventory Database

This section describes how to install and configure the Sybase ODBC driver in order to access the ZfD 3.2 Inventory database.

Sybase ODBC driver version 7.0.0.313 is available on the ZfD 3.2 *Companion* CD shipped with ZfD 3.2.

To configure Sybase ODBC driver for the ZfD 3.2 Inventory database:

1. Ensure the following prerequisites are met:
   - Windows 95/98/NT/2000 with minimum 64MB RAM.
   - 32-bit ODBC drivers version 3.5 or above.
2. Install the Sybase ODBC Driver.
   - **2a** Extract the files from \ODBC\SybaseODBC.zip from the ZfD 3.2 *Companion* CD to a drive.
   - **2b** Copy the files from the directory where you have extracted the files to \PROGRAMFILES\SYBASE directory.

     \PROGRAM FILES\SYBASE\ADAPTIVE SERVER ANYWHERE 7.0\WIN32\DBCON7.DLL

     \PROGRAM FILES\SYBASE\ADAPTIVE SERVER ANYWHERE 7.0\WIN32\DBLGEN7.DLL

     \PROGRAM FILES\SYBASE\ADAPTIVE SERVER ANYWHERE 7.0\WIN32\DBODBC7.DLL

     \PROGRAM FILES\SYBASE\ADAPTIVE SERVER ANYWHERE 7.0\WIN32\DBODTR7.DLL

     \PROGRAM FILES\SYBASE\ADAPTIVE SERVER ANYWHERE 7.0\WIN32\DBPORT6.DLL
   - **2c** Double-click the SYBASEODBC.REG file. The following message *Information in SybaseODBC.reg has been successfully entered into the registry* should be displayed.

     SYBASEODBC.REG contains the required registry settings for Sybase ODBC driver. Modifying this file may corrupt system registry settings, thus making your server unusable.
3. Configure the Sybase ODBC driver
   - **3a** From the ODBC Data Source Administrator, click Start > Settings > Control Panel > ODBC Data Sources.
   - **3b** From the ODBC data Source Administrator, select User Data Sources > Sybase ODBC > click Configure.

     This will display the ODBC Configuration for Adaptive Server Anywhere dialog.
   - **3c** Click the Login tab.

     Ensure that you have entered a user id and password as follows:

     User ID: mw_dba

Password: novell

3d Click the Database tab.

Enter the following details.

Server Name: IP address of the server on which Sybase is running.

Database Name: Database Name: mgmtdb

3e Click the Network tab > check the TCP/IP option.

3f Specify the host name or IP Address of the server.

The format should be host=*IP_address* where *IP_address* is the IP address or host name of the server on which Sybase is running.

3g Click ODBC tab > Test Connection.

The message *Connection Successful* should be displayed. This message indicates that the Sybase ODBC driver has been configured correctly. If there is an error message, repeat Step 3.

Removing the ZENworks 1.*x* Inventory Attributes from Workstations

If you are upgrading from ZENworks 1.*x* to ZfD 3.2, the 1.*x* inventory attributes are replaced in the Directory Services schema by ZfD attributes during the installation. However, the 1.*x* attributes are not removed from Directory Services. You can use the Remove ZENworks 1.*x* Inventory Utility to remove these attributes from Directory Services. Though removal of the 1.*x* inventory attributes is not necessary to run inventory in ZfD, removing the attributes will free up space for other attributes and applications in Directory Services.

To remove the ZENworks 1.x attributes:

1. Run REMINV1X.EXE.

   The REMINV1X.EXE file is found on the ZfD 3.2 *Program* CD in the \PRODUCTS\ZENWORKS\CORE\CONSOLEONE\BIN directory.

2. Type the context for the workstations for which you want to purge the 1.*x* attributes.
3. Check the Include Subcontainers check box to include workstations located in the subcontainers of the selected context.
4. Click OK to run the utility.

   A message box displays indicating that the attributes have been removed.

5. Click OK.

26 Using Workstation Inventory

This section contains the following information:

Viewing the Servers Deployed for Inventory

Using ConsoleOne®, you can view the servers and databases that you configured for collecting inventory.

To view the servers deployed for inventory:

1. In ConsoleOne, click a container > View > Complete Tree View.

   A Complete Tree View displays all the servers within the container.

2. In ConsoleOne, right-click an Inventory Service object > click View > click Up Tree View.

   An Up Tree View displays all the servers from the selected server up to the highest level (Root Server), including the database servers.

Displaying Inventory Information

The following sections will help you configure the Inventory database to display inventory information and explain the various types of information you can view:

- You can list hardware and software components found on the managed workstation and any custom information you have specified for the workstation.

  The Workstation Inventory window displays the inventory items for a managed workstation. This window displays the data from the last inventory scan for the managed workstation.

  For more information about viewing the inventory information of a managed workstation, see "Viewing the Workstation Inventory Summary of a Managed Workstation" on page 412.

- You can list managed workstations 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 workstations within the selected database sites.

  For more information about querying the Inventory database, see "Viewing Inventory Information of Managed Workstations by Querying the Database" on page 419.

- You can list minimal information stored in the eDirectory Workstation object is available.

For more information, see "Viewing the Minimal Inventory Information from an NDS Object" on page 421.

- You can use a list of reports that generate the inventory information from the Inventory database specific to your needs.

  For more information about the types of reports, see Inventory Reports.

Configuring the Inventory Database

If you want to view the inventory information of 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, click a container.
2. Click Tools > Configure DB.
3. 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. Click OK.

The database you configured is used for data retrieval unless you change it again using this same procedure.

Viewing the Workstation Inventory Summary of a Managed Workstation

ZfD provides the following inventory information collected from the managed workstations:

Scan Data Group	Scan Data Item	Description
HardwareSoftware Inventory > General	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 workstation
	ModelNumber	Serial number value for the computer, assigned during manufacture
	Primary Owner Contact	The phone number of the primary user of this system
	Primary Owner Name	The name of the primary user or owner of this system
HardwareSoftware Inventory > General > Login Details	Current login user	User logged in to the Primary Novell® eDirectory™ tree when the workstation was scanned
	Last login user	User logged in last to the Primary eDirectory tree when the workstation was scanned

Scan Data Group	Scan Data Item	Description
Software > Software Vendors	Software Vendor Name	Name of the software manufacturer
	Name	Name of the software application
	Version	Version of the software
Software > Operating System	Code Page	Language Code Page of the operating system
	Description	Operating system description
	Install Date	Install date of the operating system
	Name	Operating system name. For example, Windows* 95/ Windows 2000
	Size Stored in Paging Files	Page file size
	Version	Version of the operating system
Software > Inventory Scan Information	Inventory Server	Name of the inventory server to which the scans are sent
	Version	Version of the Scanner
Hardware > Mouse Driver	Name	Name of the mouse driver
	Version	Version number of the mouse driver
Hardware > Modem	Caption	Modem label
	Description	Additional information about the modem
	Name	Identifying information of the modem
Hardware > Processor	Device ID	Special hexadecimal string identifying the processor type
	Current Clock Speed (in MHz)	Current clock speed of the processor
	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 > BIOS	BIOS Identification Bytes	Byte in the BIOS that indicates the computer model

Scan Data Group	Scan Data Item	Description
	Install Date	The manufacturing date of the BIOS
	Manufacturer	BIOS vendor name
	Name	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 > Bus	Protocol Supported	Bus protocol indicates PCI, ISA, and others
	Version	Version of the bus supported by the motherboard
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 > Display	Current Bits Per 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
	Description	Description of the monitor
	Max Memory Supported	Maximum memory that the display adapter supports for VIDEO RAM
	Max. Refresh Rate	Maximum refresh rate of the monitor for redrawing the display, measured in Hertz
	Min. 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
	Video Architecture	The architecture of the video subsystem in this system, for example, CGA/VGA/SVGA/8514A
	Video Memory Type	The type of video memory for this adapter, for example, VRAM/SRAM/DRAM/EDO RAM
Hardware > Adapter	Description	Adapter description

Scan Data Group	Scan Data Item	Description
Hardware > Display > Video Driver	Install Date	The Video BIOS release date
	Is Shadowed (True or False)	If True, the Video BIOS is currently being shadowed
	Manufacturer	Vendor name
	Version	The version number or version string of the Video BIOS
Hardware > Ports > Parallel Ports	Parallel Port Name	The logical name of the input-output device on this parallel port, under this operating environment
	DMA Support (True or False)	If True, DMA is supported
	Address	Base I/O address for this parallel port
	IRQ	IRQ number of the parallel port
Hardware > Ports > Serial Ports	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 > Disk > Floppy > Floppy Drive	Drive Letter	Mapped drive name of the floppy drive
Hardware > Disk > Floppy > Floppy Disk	Capacity	Floppy drive capacity
	Description	Floppy drive description
	Manufacturer	Vendor name
	Physical Cylinders	Floppy drive cylinders
	Physical Heads	Floppy drive R/W heads
	Sectors/Track	Floppy drive sectors per track
Hardware > Hard Disk	Capacity	Hard disk capacity
	Description	Description
	Manufacturer	Vendor name
	Physical Cylinders	Cylinders
	Physical Heads	Heads
	Sectors/Track	Hard disk drive sectors per track
Hardware > Hard Disk Logical Drive	Caption	Hard disk volume label
	Drive Letter	Letter name of the drive
	Size	Drive's actual size in MB

Scan Data Group	Scan Data Item	Description
	Free Disk Space	Drive's available space in MB
	Volume Serial Number	Hard disk volume serial number
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 > Memory	Total Virtual Memory Size	Indicates the total number of bytes in the virtual address space of the calling process
	Physical Memory Size	Total physical memory of the workstation
Hardware > Sound Card > Sound Adapter	Description	Description of the multimedia component for the workstation
	Manufacturer	Vendor name
Hardware > Multimedia Card	Description	Description of the multimedia component for the workstation
	Manufacturer	Vendor name
	Name	Label of Multimedia card
Hardware > Backup Disk	Description	Description of the backup device for the workstation
	Name	Name of the backup device vendor
Hardware > Backup Disk > Logical	Drive Letter	Name of the mapped drive to the backup device
Hardware > Inventory	Physical Memory (size in KB)	Total physical memory of the workstation
Hardware Software Inventory	Primary Owner Contact	Phone number of the primary user of this system
	Primary Owner Name	Name of the primary user or owner of this system
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 > Mouse	IRQ Number	Interrupt assigned to this device

Scan Data Group	Scan Data Item	Description
	Name	Identifying information of the mouse
	Number of Buttons	Number of buttons on the mouse
Hardware > CDROM	Caption	CD-ROM Label
	Description	Description of the CD-ROM
	Manufacturer	Vendor Name
Hardware > CD ROM > Logical Drives	Drive Letter	Drive letter used for CD drives
Hardware > Local File System	File System Type	File system associated with this partition, for example, FAT/ NFS
	Name	Name that the system uses to identify the partition, usually this is the drive letter
	Size	Size of the partition
Hardware > Network Adapter Driver	Description	Description of the network adapter driver
	Name	Name of the network adapter driver
	Version	Version of the network adapter driver
Hardware > Network Adapter	Max. Speed	Rate at which the data is transferred over the LAN
	Name	Network adapter name
	Permanent Address	Node address stored permanently in the adapter
Network	IP Address	The unique address assigned to a computer on an IP Internet
	Subnet Mask	The subnet mask of the workstation 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
	DNS Name	The DNS name of the workstation
	IPX Address	The IPX™ address of the workstation
	MAC Address	Unique node address permanently coded in the network adapter that identifies a specific computer on a network
Network	NetWare® Client Version	Version of the Novell Client™ software installed on the workstation
System Information > 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

Scan Data Group	Scan Data Item	Description
	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
System Information > Motherboard	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
System Information > System IRQ	IRQ Number	Number of the Interrupt Request Line (IRQ), from 0 to 15
System > System DMA	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
	IRQ Availability	Indicates whether Virtual Direct Memory Access (DMA) is supported
System > Slot	Description	Card currently occupying this slot
	Max Data Width	Maximum bus width of cards accepted in the slot
	Thermal Rating	Maximum thermal dissipation of the slot in milliwatts

This window displays the data from the last inventory scan for the managed station.

For more information about the DMI/WMI data display, see "Viewing the DMI/WMI Data in the Inventory Summary Information" on page 419.

To view the inventory information of a managed workstation:

1. In ConsoleOne, click Tools > Configure DB.

   For more information, see "Configuring the Inventory Database" on page 412.

2. Right-click a managed workstation > click Actions > click Inventory.

Viewing the DMI/WMI Data in the Inventory Summary Information

The following list contains the DMI/WMI components that are scanned if they are instrumented and available in the DMI/WMI database on the workstation:

- BIOS Characteristics
- Disks
- DMA
- IRQ
- Keyboard
- Network Adapter Driver
- Network Adapter Hardware
- Operating System
- Parallel Ports
- Pointing Device
- Power Supply
- Processor
- Serial Ports
- SubComponent Software
- System Cache
- System Contact Information
- System Enclosure
- System Hardware Security
- System Memory Settings
- System Power Controls
- System Resource DMA Info
- System Slot
- Temperature Probe
- Video
- Video BIOS
- Video Output Device
- Voltage Probe

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

Viewing Inventory Information of Managed Workstations by Querying the Database

Using ConsoleOne, you can query the Inventory database to display the hardware and software components of managed workstations that you want to view. The Inventory Query window displays the information satisfying the criteria you specify.

The Inventory database stores inventory data (hardware, memory, and environmental information) for each managed workstation. 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.

NOTE: If you are generating inventory reports from an Inventory database that has inventory information of more than 2,000 workstations, we recommend that you use the Data Export tool for better performance. The Data Export tool generates the inventory information in to a comma separated value (.CSV) file. Use this .CSV file along with any standard reporting tool such as Seagate* Crystal Reports.

To query the Inventory database for inventory information:

1. In ConsoleOne, click a container.
2. Click Tools > Configure DB.

   For more information, see "Configuring the Inventory Database" on page 412.
3. Click Tools > Inventory Query.
4. Specify the criteria for query:

   Find In: Choose a site name from the list. The query locates all workstations within the specified site satisfying the query expression.

   Search Entire Database: Check this option to include all database sites while querying. The query locates all workstations in all sites 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. Combine multiple query groups in which 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 workstations have a VGA color video adapter.

   Attributes: Select the component attributes. Attributes that you can specify to query the managed workstations are as follows:

   - Software (Name, Vendor, Version)
   - Operating System (InstallDate, Name, Version, Code Page)
   - Scanner Information (Version, LastScanDate, NumberofFilesScanned)
   - NetWare Client details
   - Hardware Information (mouse, keyboard, video BIOS, display adapter, BIOS, processor, memory (virtual, physical), disk, floppy drives, hard disk, CD-ROM drive, backup disk drives, ports (serial ports, parallel port), bus, modem, sound card, battery, power supply, cache)
   - Network (IP address, IPX address, MAC address, DNS)
   - System (Motherboard, IRQ, Slot, DMA)

   Operator: Form query groups that will be combined with the previous query group by using the relational operator specified between the query groups.

   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. Use the wildcard character % to substitute any number of characters, or the ? character to substitute one character in the Value field.

The list of description values displayed for an Inventory component is taken from the Inventory database corresponding to the component.

Save and Load the Existing Query Files: Save the queries in a file and load the query file as required.

5 Click Find.

This will query based on the query criteria you specify and display the workstations that match the query in the Query Results window.

In the Query Results window, double-click the workstation to view the inventory information of the workstation.

Viewing the Minimal Inventory Information from an NDS Object

The scanners store minimal scan data as an eDirectory Workstation object. You can view this minimal information from ConsoleOne. The Minimal Information page lists the inventory information of the scanned managed workstations.

For more information about the listed items in this page, see "Mapping Between Minimal Information Attributes and Attributes in the Inventory Database" on page 421.

To view the inventory information stored in eDirectory:

1 Right-click a managed workstation that has been successfully scanned > click Properties > click the ZENworks Inventory tab > click Minimal Information.

If you click More Workstation Information in this page, the Workstation Inventory window will be displayed.

Mapping Between Minimal Information Attributes and Attributes in the Inventory Database

The following table shows the mapping between minimal information attributes and attributes in the Inventory database:

Minimal Information	Inventory Database
Asset Tag	Scanned Inventory - Asset Tag
BIOS Type	Hardware - BIOS - Type
Computer Model	Scanned Inventory.Computer Model
Computer Type	Scanned Inventory.Computer Type
Disk Information	Hardware - Disk - Hard Disk - Drive Letter
IP Address	Network - IP Address
IPX Address	Network - IPX Address
Last Scan Date	Date and time when the workstation was scanned
MAC Address	Network - MAC Address
Memory Size	Hardware - Memory - Total Extended Memory
Model Number	Scanned Inventory.Model Number

Minimal Information	Inventory Database
NIC Type	Hardware - Network Adapter Driver - Description
Novell Client	Novell Client version
OS Type	Software - Operating Systems - OS - Name
OS.Version	Software - Operating Systems - OS - Version
Processor	Hardware - Processor
Serial Number	Workstation Serial Number
Subnet Mask	Network - Subnet Mask
Video Type	Hardware - Display - Type

Running Inventory Reports

You can run reports to gather inventory information from the Inventory database.

The inventory information is taken from the Inventory database you configure. For more information, see "Configuring the Inventory Database" on page 412.

Once you have configured the database, you access the hardware and software reports through the Tools menu from ConsoleOne. You can select from a predefined set of report forms to generate a report. The inventory report is displayed in the viewer window.

You can print or export the report as desired. Remember that any reports you generate will be empty if you haven't configured ZfD to start populating the Inventory database with the data you want.

Before running the inventory reports, you must configure the inventory database. See "Configuring the Inventory Database" on page 412. The inventory reports always use the Inventory database you configured as the data source for your reports unless you change it later as described in "Configuring the Inventory Database" on page 412.

To generate the inventory report:

1. In ConsoleOne, click a server object.
2. Click Tools > Inventory Reports.
3. 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.
4. Specify the selection criteria.

   For example, if you want to view the inventory information of all database sites, select the report type you want and specify % as the Database Site Name. The report will display the inventory information of any database sites within the configured Inventory database. You can also specify the database site that you want.

   Depending on the type of report you want, you can filter the information. For example, to view all workstations of a particular database site with the Windows NT operating system, you

select the Workstation Operating System Listing and specify the selection criteria Operating System Type as Windows NT, Operating System Version as 3.0, and the database site.

See the table with selection criteria for the Inventory Reports.

5 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.

The following table lists the selection criteria for the inventory reports.

Report Name	Selection Criteria
Workstation Scan Time Listing	Database Site Name
Workstation Operating System Listing	Operating System Type, Operating System Version, Database Site Name
Workstation BIOS Listing	BIOS Install Date, Database Site Name
Workstation Processor Listing	Processor Family, Database Site Name
Workstation Processor Speed Listing	Lower Bound of Processor (in MHz), Upper Bound of Processor (in MHz), Database Site Name **NOTE:** Ensure that the value for Lower Bound of Processor is less than the value for Upper Bound of Processor.
Workstation Video Adapter Listing	Video Architecture, Database Site Name
Workstation Network Adapter Listing	Network Adapter Name, Database Site Name
Workstation Software Listing	Software Name, Software Version, Database Site Name
Workstation Memory Listing	Lower Bound of Blocks of Memory (in MB), Upper Bound of Blocks of Memory (in MB), Database Site Name **NOTE:** Ensure that the value for Lower Bound of Blocks of Memory is less than the value for Upper Bound of Blocks of Memory.
Software Summary Listing	Software Name, Software Version, Database Site Name
General Workstation Inventory Report	DN of the workstation, Database Site Name
Asset Management Report	DN of the workstation, Database Site Name
Hardware Inventory Report	DN of the workstation, Database Site Name
Networking Information Report	DN of the workstation, Database Site Name
Software Inventory Report	DN of the workstation, Database Site Name
Software Inventory Report for the Entire Site	Database Site Name

If the Reporting dialog box allows wildcards, you can use an asterisk (*) and question mark (?), or their SQL equivalents, percent (%) and underscore (_). The wildcard characters can be used for character data only.

The following table lists examples of wildcards.

Example	Specifies to Include
%	All items
wNT%	All items starting with "wNT"
wNT_cpq	All items starting with "wNT" followed by any character and "cpq"
wNTcpq.xcorp	The single named item, in this case a workstation

You can generate the types of reports described below, assuming you have already configured ZfD to start populating the inventory database with the data you want. The following table lists the Simple Inventory lists that provide information on individual aspects of workstation inventory, such as operating system.

Report Name	Information Provided
Workstation Scan Time Listing	Date and time of the last inventory scan on each workstation
Workstation Operating System Listing	List of all the workstations with an OS type, an OS version, and the total number of such workstations
Workstation BIOS Listing	List of all the workstations with a BIOS release date, and the total number of such workstations
Workstation Processor Family Listing	List of all the workstations with a processor family (such as Pentium Pro), and the total number of such workstations
Workstation Processor Speed Listing	List of all the workstations within a range of processor speed (such as 200-400 MHz), and the total number of such workstations
Workstation Video Adapter Listing	List of all the workstations with a video adapter (such as MGA 2064W), and the total number of such workstations
Workstation Network Adapter Listing	List of all the workstations with a network adapter (such as 3Com* Fast EtherLink*) and the total number of such workstations
Workstation Software Listing	List of all the workstations with a software name, version, and the total number of such workstations
Workstation Memory Listing	List of all the workstations within a range of memory, and the total number of such workstations
Software Summary Listing	Lists the total number of workstations with a particular software and version

The following table lists the Comprehensive Inventory Reports that combine several aspects of workstation inventory into each report, such as memory, hard disk, and processor.

Report Name	Information Provided
General Workstation Inventory Report	BIOS, computer description, OS description, display details, NIC, physical disk drive, IP address, and MAC address for each workstation
Asset Management Report	BIOS, computer description, processor, and OS description for each workstation
Hardware Inventory Report	Memory, processor, display details, physical disk drive, and modem for each workstation
Networking Information Report	OS Description, MAC address, NIC, and IP description for each workstation
Software Inventory Report	Software with product name, version, and vendor for each workstation
Software Inventory Report for the Entire Site	All softwares with product name, version, and vendor for the entire database site

To print a report:

1. Generate and view the report.
2. On the toolbar, click the printer icon.
3. In the Print dialog box, select the print options you want > click OK.
4. In the next Print dialog box, click OK to print in the default orientation mode.

   To change the mode, click Properties > click Orientation > select the mode (Landscape/ Portrait) > click OK twice.

The Inventory Lists are designed to be printed in portrait mode; all the other reports are designed to be printed in landscape mode.

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 dialog box, specify the location and file format > click OK.

Exporting the Inventory Data to CSV Format

You can customize the inventory data you want to export from the ZfD 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 workstations whose attributes will be exported. For example, you can export only those workstations with a particular processor speed. The Data Export tool will export all workstations 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

To invoke the Data Export tool:

1. In ConsoleOne, click a container.
2. Click Tools > click Configure DB > select a ZENworks Database object. > click OK.
3. Click Tools > Data Export.

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 426.
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 workstations and the database sites 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. Select the database fields from the list of Database Fields > 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.
4. Specify the filter conditions for workstations > click Edit Query. See "Forming the Query and Setting the Filter Conditions" on page 427.

   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 > click Save.

   The configuration file (.EXP) contains the settings such as the inventory components you selected, and also the query formed for filtering the workstation 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 Click 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.

To run the data export tool from a server, click Perform the Query on a Remote Server. See "Running the Data Export Program From the Server" on page 429.

7 Click Finish.

8 Specify the .CSV filename > click OK.

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.

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 426.

2 Select Create a New Database Query.

3 Set the following filter conditions.

Selecting the database sites: In the Find In option, select the database sites names from the list.

The Data Export tool will export the data from the Inventory database sites you select. This tool locates all workstations satisfying the filter conditions within the selected database sites.

Click the Browse button to select one or more database site names from the list. In the Select Sites window, click one or more database sites. The selected site names are shown in double quotes.

To include all database sites for data export, check the Search Entire Database check box. The data export tool will locate those workstations satisfying the filter conditions in all database sites.

Selecting the attributes of the Inventory Components: Click the Browse Attribute button to select component attributes in the Select Attribute window. 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.

Components are as follows: General, Software, Hardware, Network System.

Operator: Relational operators show the relationship between the component and the value. Use the Matches option to specify the wildcard characters in the Value field.

Specifying the 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. Use the wildcard character % to substitute any number of characters, or the ? character to substitute one character in the Value filed.

The list of description values displayed for an Inventory component is taken from the Inventory database corresponding to the component.

Specifying the 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 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 workstation 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 426. 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 426.

3 Select Open a Saved Database Query > click Next.

The default directory for .EXP files is CONSOLEONE\version\REPORTING\EXPORT. Click Browse to open an existing .EXP file.

If the database setting specified in the .EXP file are not valid, the data export tool uses the default setting, which includes all databases for data 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 Click 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 a .CSV file.

To run the data export tool from a server, click Perform the Query on a Remote Server. See "Running the Data Export Program From the Server" on page 429.

Click Finish.

7 Specify the .CSV filename > click OK.

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.

Running the Data Export Program From the Server

Running the Data Export program from a server is recommended if you are exporting data from a large database with more than 10,000 workstations or if you have specified complex queries with more than 20 database fields selected for exporting.

To run the data export program from the server:

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

   Follow the Step 1 to Step 5 outlined in "Exporting the Inventory Data to a CSV File" on page 426 and ensure that you save the settings in the .EXP file.

   When you save an EXP file, a corresponding data configuration file is created in the same directory with the same filename as the .EXP file and with the .CFG file extension.

2. Click Perform the Query on a Remote Server to run the data export program from any server that has Workstation Inventory components installed > click Finish.

3. Copy the .EXP file and .CFG file to the server.

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

   From the 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.

4. Choose the Inventory sites.

   - To export the data from all database sites, satisfying the filter conditions, type 0.
   - To choose the database sites, type the numbers corresponding to the site names in the displayed list.
   - To select multiple site databases, separate the site numbers corresponding to the site names by commas.

   The data export displays the number of workstations 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.

27 Troubleshooting Workstation Inventory with Status Logs

Novell® ZENworks® for Desktops lets you track whether the scan or the roll-up of information is successful by viewing the log files and scan history.

The inventory components report the status of the inventory scanning and roll-up of scan information in Novell eDirectory™.

For example, when you view the scan logs, you can determine whether the scan was successful or if there were any errors while scanning the workstation or at the time of roll-up.

You can view the following status information:

Viewing the Scan History of a Workstation

The Scan Status reports the history of the scans done at the workstation. For example, you view the Scan Status window to determine whether the scan was successful for the workstation, or whether the Storer has stored the .STR files of the workstation in the database. The inventory components (Scanner and Storer) write the scan information in the Status log.

The Scan Status displays the history of the latest ten scans of the selected workstation.

The following table lists the details of the log:

Status Information	Details
Time of Scan	Displays the date and time of the scan.
Message	Displays the message reported by the inventory components while scanning the workstation and storing the scan data in the database. For more information, see Workstation Inventory Error Messages in *Error Messages*.

You can export the log file as a .CSV or TAB-delimited file.

To invoke the Scan Status window:

1 In ConsoleOne®, right-click the workstation > click Properties > click the ZENworks Inventory tab > click Scan Status.

If the scan is disabled in the Inventory policy, the Scanner does not log any status messages in the Status Report log.

Viewing the Roll-Up History of the Server

The Roll-Up Status reports the status of the roll-up information from the 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 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 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 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 servers. For more information, see Workstation Inventory Error Messages in *Error Messages*.

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 > click Roll-Up Status.

Viewing the Status of Inventory Components on a Server

The Server Status window reports the status of the inventory server components on the selected server. You can view the 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, Service Manager, Roll-Up Scheduler) are not up and running on the server, the status of the 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.
Message	Displays the message reported by the inventory components. For more information, see Workstation Inventory Error Messages in *Error Messages*.

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 > click Server Status.

Viewing the Status of the Last Scan in the Workstation Scan Log

The Workstation Scan log reports the status of the latest scan done at the workstations associated with the selected Container. For example, you view the Workstation Scan log to determine whether the latest scans were successful for the workstations or whether the Storer has stored the workstation .STR files in the database. The inventory components (Scanner and Storer) write the scan information in the Status log.

You can also choose whether to display error, warning, and informational status messages of the workstations.

The following table lists the details of the log:

Status Information	Details
Scanned Workstation Name	Displays the DN of the workstation.
Time of Scan	Displays the date and time the status was logged.
Message Type	Displays the severity of the message.
Message	Displays the message reported by the inventory components while scanning the workstation or storing the inventory information in the database. For more information, see Workstation Inventory Error Messages in *Error Messages*.

You can export the file as a .CSV or TAB-delimited file.

NOTE: If the scan is disabled in the Inventory policy, the Scanner does not log any status messages in the Status Report log.

To view the Workstation Scan log window:

1. In ConsoleOne, click the container > Tools > Workstation Scan Log.
2. Click the message severity type > OK.

When the Scanner creates a .STR file of a workstation and the Selector processes this file, the inventory components also log the status of the scanning in the Workstation Status log. It is possible that at the same time, the Storer is processing the .STR files of another workstation. Meanwhile, the Scanner continues scanning and updates the Workstation Status log for the next scan. The Selector merges the .STR files of the previous scan and the current scan. In this scenario, the Storer processes the merged .STR files and updates the Workstation Scan log for the corresponding last scan.

Viewing the Roll-Up Log for 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 workstation. For more information, see Workstation Inventory Error Messages in *Error Messages*.

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 > Roll-Up Log.
2. Click the severity type of the messages you want to view > 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 > 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 in ConsoleOne
Workstation Scan Log	Scan program, Storer	Scanned workstation name, time of scan, inventory component, message type, and status message	Click the container > Tools > Workstation Scan Log
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 > Roll-Up Log
Workstation Scan Status	Scan program, Storer	Time of scan and status message	In ConsoleOne, right-click the workstation > click Properties > the ZENworks Inventory tab > Scan Status
Status of Inventory components on Server	Sender, Receiver, 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 SYS:\PUBLIC\ZENWORKS on the server.

The INVENTORYLOG.XML log file is located in the SYS:\PUBLIC\ZENWORKS\WMINV\LOGS directory on NetWare® and Windows* NT*/2000 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 servers, this file is in the SYS:\PUBLIC\ZENWORKS directory.

The contents of INVENTORYLOG.INI is 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.

If you want to view the XML log file in languages other than English, see Unable to view the inventory status reports (XML format) in languages other than English in Workstation Inventory in *Troubleshooting*.

C Documentation Updates

This section lists updates to the Workstation Inventory part of the *Administration* guide that have been made since the initial release of Novell® ZENworks® for Desktops (ZfD) 3.2. The information will help you to keep current on documentation updates and, in some cases, software updates (such as a Support Pack release).

The information is grouped according to the date when the *Administration* guide was updated and republished:

October 31, 2002 (ZfD 3.2 Support Pack 1)

Location	Update
"WMI-Compliant Scanners" on page 298	Updated the link to the Microsoft WMI Web site.

October 10, 2003

Location	Update
"Running the Data Export Program From the Server" on page 429	Updated Step 3 on page 429.