

HP Discovery and Dependency Mapping Inventory

for the Windows® operating system

Software Version: 7.60

Reference Guide

Manufacturing Part Number: None
Document Release Date: August 2009
Software Release Date: August 2009



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 1993-2009 Hewlett-Packard Development Company, L.P.

Trademark Notices

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

Windows™ Vista is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Java™ is a US trademark of Sun Microsystems, Inc.

AMD® is a trademark of Advanced Micro Devices, Inc.

UNIX® is a registered trademark of The Open Group.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Support

You can visit the HP software support web site at:

www.hp.com/go/hpsoftwaresupport

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract. To find more information about support access levels, go to the following URL:

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

To register for an HP Passport ID, go to the following URL:

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

Contents

1	Introduction	9
2	How DDM Inventory Works	11
	Introduction	11
	What is Discovery?	11
	Network Explorer	13
	Update Network Model	13
	Model	14
	Updating the Device Model for a Single Device	15
	Updating Device Models for Multiple Devices	15
	Scanning Devices for Inventory Data	16
	Scanning Devices for Application Utilization Data	16
	Filter Out of Device Database	17
	Add to Device Database	17
	Rulebase	18
	Data Going into the Rulebase	18
	Creating a Device Type	19
	Adding Rules to the Rulebase	19
	Printers	21
	Table Reader	21
	Generate Add Event	21
	Choose a Scanning Method	21
	What is an Agent?	22
	Agent-Based and Agentless Scanning	22
	Securing Agent Communication	22
	Further information	23
	Schedule Scan	23
	Automatic Deployment of Scanners	23
	Scheduling of Scan Execution	23
	Collection and Storage of Scan Files	23
	Scan File Enrichment	24
	Poll Device	24
	How Long Does it All Take?	25
	DDM Inventory is Always Discovering	25
	DDM Inventory's Topology Map	26
	How Much Network Bandwidth Does the Server Need?	26
	Discovery Ping Rate	27
	Table Reading and Polling	27
	Scanning/Agent Deployment	28
	Benchmarking Data	28

WAN	30
Aggregating DDM Inventory servers	32
What Is an Aggregator?	32
How Do I Use the Aggregator?	32
Aggregator Data Transfer	33
Connecting the Reports Database on the Aggregator Server	38
The Optimum Time to Run a Connect-It Scenario	38
RFCs Supported by DDM Inventory	38
Communication Models	40
Frame Relay	40
FDDI	42
HSRP	43
Scheduled Events	43
Software Library	44
Knowledge Updates	44
3 Terms and Concepts	45
Network Terms and Concepts	46
SNMP	46
MIB	46
Domain Names	46
Address Types	46
Netmask Notation	48
Community Strings/Users	48
Bridge Aging	49
OSI Model Layers	50
Management Workstation	50
Virtual Machine	50
Solaris Zone	51
DDM Inventory Terms and Concepts	52
Object Label	52
Events and Alarms	53
Panel Elements	54
4 Virtualization in DDM Inventory	57
Overview	57
Supported Virtualization Technologies	58
VMware	58
Solaris Zones	59
Support for Virtual Devices in DDM Inventory	60
Discovering Virtual Devices on the Network	60
Scanning Virtual Devices	60
Virtual Devices in the DDM Inventory Database	66
Virtual Devices on the Network Map	66
Virtual Devices Window	67
Virtual Devices and the Device Manager	67
Virtualization Configuration Profiles	68

Reports on Virtual Devices	68
Deactivation and Purging of Virtual Devices	68
5 Mobile Devices in DDM Inventory	71
Integration with Other HP Software Products	72
Overview	72
Mobile Configuration Profiles	74
Enabling and Disabling Mobile Discovery	74
Using a Secure Connection for Mobile Discovery and Inventory	75
Mobile Devices in the DDM Inventory Database	75
Protecting Private Information for Mobile Devices	75
Mobile Devices on the Network Map	76
Scanning Mobile Devices	76
Finding Mobile Devices	76
Mobile Devices and the Device Manager	77
Mobile Device Servers	77
Individual Mobile Devices	79
Reports	79
Deactivating and Purging Mobile Devices	79
Viewing Your Current Mobile Profile Settings	80
6 Recorded Events	81
Port Adds and Deletes	81
Port Changes	81
Device Adds and Deletes	81
Device Changes	81
Exceptions	81
Not Recently Seen	81
7 Scanners	83
Scanner Types	84
Viewing the Results of the Scan	85
Command Line Parameters and Switches	86
How to Use a Command Line Parameter	86
Command Line Parameters for Scanners	87
Viewing Command Line Options in Viewer or Analysis Workbench	92
Using Command Line Switches to Enable and Disable Specific Hardware Tests	92
Starting the Scanners	94
Information Collected by the Scanners	94
Starting the Scanner Manually	94
Hardware Scan	95
Software Scan	95
Scanner Error Level Codes	95
MSI Scanner	97
Overview of the MSI Scanner	97
Starting the MSI Scanner	97
Opening the MSI Scanner Output File in the MSI Importer	97

MSI Scanner Error Level Codes	98
Troubleshooting	99
Using Error Level codes	99
Scanner Generator Errors	99
Hardware Scanning Errors	99
Software Scanning Errors	100
Scan File Saving Errors	100
Additional Errors	100
Using Scanners for Manual Inventories	102
Walkround Inventory	102
Using a Distribution Tool	102
Command Line Execution	102
8 Logging User Actions	103
Audit Log	104
Discovery Log	105
9 UI Shortcuts	107
Major Components	107
Asset Questionnaire	108
Device Manager	108
Port Manager	109
Line Manager	110
Attribute Manager	111
Service Analyzer	112
Index	113

1 Introduction

This guide contains several chapters that you may find useful as you use DDM Inventory. Included are many definitions of terms and concepts used in DDM Inventory, as well as other reference materials that might help you work with the product.

2 How DDM Inventory Works

This section explains the basic functions of DDM Inventory. If you are a new user, we recommend reading part of it to help orient yourself with the product and allow you to have a better experience using DDM Inventory.

You can use DDM Inventory without ever having to read or refer to this section of the manual. However, even experienced DDM Inventory Administrators are likely to find it easier to understand the behavior of DDM Inventory after reading this section.

Introduction

DDM Inventory will provide you with an enormous amount of data about your network and devices—including virtual devices and mobile devices. It can discover devices on its own by pinging and polling through a collection of IP addresses (device groups) that you provide, and it can also collect data from devices using scanners that you can customize.

With a full install of DDM Inventory, you will receive many tools to help collect, analyze, and report on your network devices. This chapter will explain some of the concepts that will help you understand DDM Inventory, and to help you maximize your benefit from using DDM Inventory.

What is Discovery?

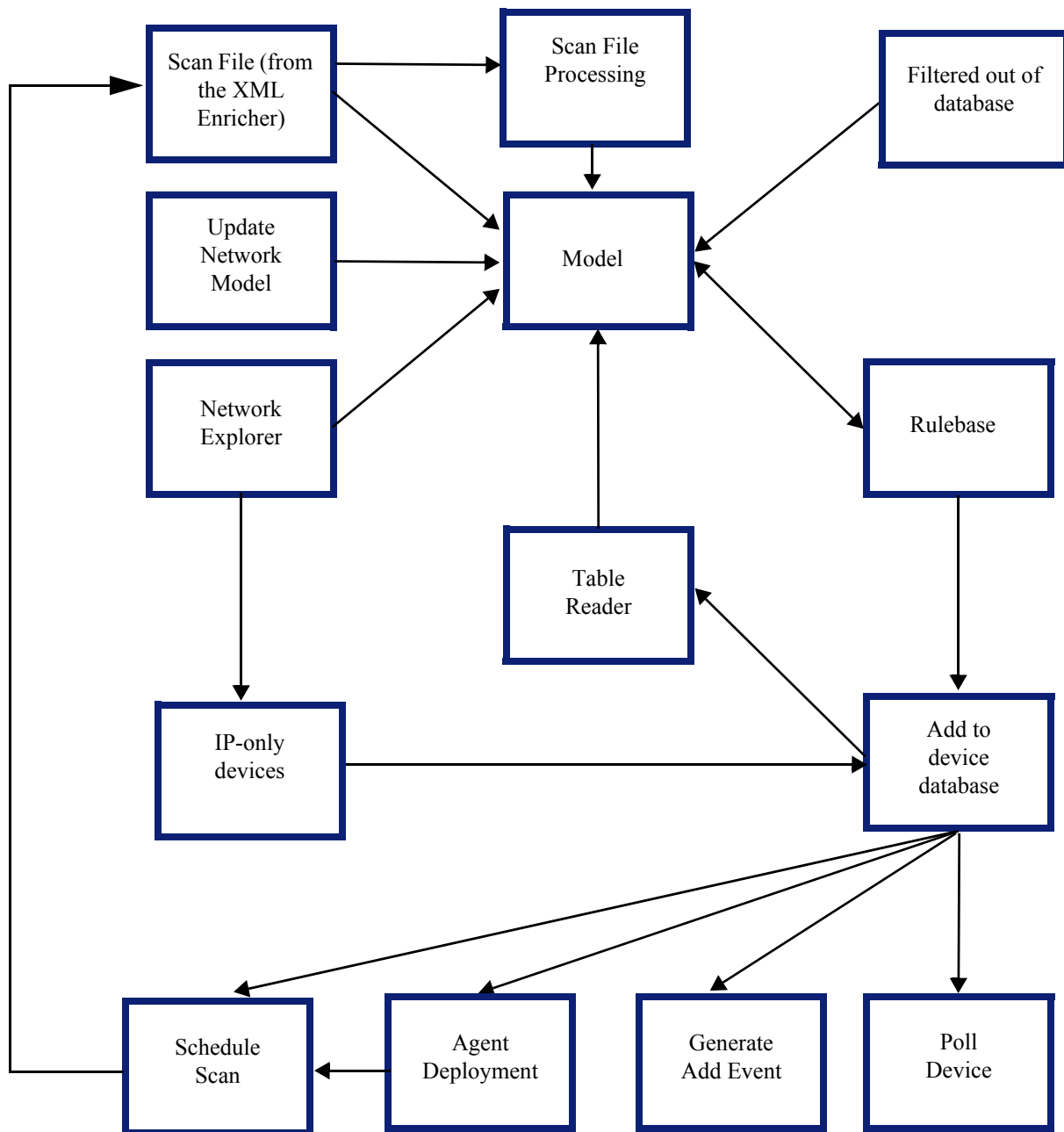
In order for DDM Inventory to provide you with network and device data, it must first know what devices are in your network, and what kind of data you want to see.

This process—discovering the network devices—is called Discovery.

Once a device has been discovered, DDM Inventory creates a device model (a unique description of the device) and adds that device model to the DDM Inventory database.

Rules from a Rulebase are run against the model, and some characteristics of the devices are determined based on the rule matches.

This flow chart shows a basic representation of how devices are discovered by DDM Inventory. Each section of the chart will be described below.



Network Explorer

Discovery is strictly a yes-or-no proposition. The Network Explorer goes through each IP address in random order and pings it once to see whether or not there is a response. Is there a device at this address or not? When there is a positive response, the IP address is added to the database (if it is not already in the database) and is sent to the modeler.



By default, the Network Explorer will ping each address once. You can change the number of pings at **Administration > System Configuration > Network devices > Number of Explorer pings per device**.

This means that the average time to discover a device can be calculated as the number of IP addresses in a device group divided by the ping rate.

The Explorer works continuously to find any new devices in the configured device groups. For faster discovery, the Explorer also tracks devices that have responded positively and omits them the next time.

Update Network Model

The Update Network Model feature can be used in the following cases:

- To force the discovery of a new device that DDM Inventory has not found on its own. The Find command should be used to enter the IP address of a new device and the Update Model command should be issued from the right-click menu.
- To update a model for an existing device. This is helpful if you have made any physical changes to a device and want these changes to be picked up as soon as possible. The Find command can be used to locate the device and the Update Model command can be issued from the right-click menu.

Once the Device Manager for the device is opened, click the **Update Model** button and select **Query Device**. This command requests that the device have its network model updated immediately.

You can also update a model in **Administration > Data management > Update network model**. This command is especially useful in the case when an existing device has changed its IP address and the network model has not yet been updated to reflect this.



The IP address used for model updates via the Device Manager is the primary IP of the device. If you want to use a different IP address use the **Administration > Data management > Update network model**.

There are two basic ways for DDM Inventory to discover devices:

Method	Explanation
Network Explorer	<p>The Network Explorer is a component of DDM Inventory that pings your network looking for devices.</p> <p>You can configure the Network Explorer by setting up device groups (which control <i>where</i> it looks) and by setting certain preferences like the Explorer Ping Rate (which control <i>how</i> it looks).</p> <p>When you prepare DDM Inventory for exploration, the Network Explorer begins by exploring the device groups you have set up for “Active discovery” in Administration > Discovery Configuration.</p> <p>Whenever a device is found in the range (or ranges) defined by your device groups, DDM Inventory will add that device to the database.</p>
Update Network Model	<p>The Update Network Model feature allows you to add a network device to the database before it has been discovered by the Explorer or scanned with a Scanner.</p> <p>This is a manual process, usually done by using the Find command to enter the IP address of a new device. A warning is shown to indicate that the device is not available in the database, but the link to the new device appears. You can double-click the device name to open the device manager.</p> <p>Alternatively, you can select the Update Model command from the right-click menu of the device name.</p>

Model

Every device has a “device model” in the DDM Inventory database.

For mobile devices, the primary source of data is the mobile device server (see [Mobile Devices in DDM Inventory](#) on page 71). For virtual devices, the primary source of data is the physical host server (see [Virtualization in DDM Inventory](#) on page 57).

For other types of devices, the two primary sources of data for the device model are the device MIB and the inventory scan file—the combination of which provides very good coverage of the infrastructure. The MIB is retrieved through SNMP and is normally available for core network devices such as routers, switches and bridges as well as network printers, etc. The scan file is generated by the scanner. Scanners are supported for a wide range of desktop and server systems.

DDM Inventory attempts to find out the device’s community strings, its domain name, NetBIOS name, how many ports each device has, and what type of device it is—whether it supports bridge tables, arp tables, Cisco CDP, source address capture, and so on.

DDM Inventory will supplement that information with data from ARP caches from other devices (routers) in the network for unmanaged devices or if you have enabled the “accumulate IP addresses” in **Administration > Discovery Configuration**.

In some cases information from one device’s MIB is used to describe another device. For example if two Cisco devices are directly connected, the CDP information from one gives you information (like SysName, SysDescr, etc.) for the second one, even if the second one is not SNMP managed.

Updating the Device Model for a Single Device

If you have Administrator or IT Manager privileges, you can use the Update Model feature in the Device Manager to update the device model for a single device. See the [Update Model \(Administrator or IT Manager\)](#) section in [Chapter 8, Using the Device Manager](#) in the *Network Data Analysis Guide* for more information.

Alternatively, you can use the **Administration > Data Management > Update network model** to update the device model for a single device.

Updating Device Models for Multiple Devices

Provided that you have Administrator or IT Manager privileges, you can also update device models for a group of devices at one time. That group can consist of all the devices in a single device group, all devices that share a particular configuration profile, or all devices in the DDM Inventory database.

To access the multiple update feature, select **Administration > Data Management > Update device models**.

To perform a multiple update, you must first specify an action to perform. The following actions are available:

- Query Device
- Run VMware Discovery
- Run Mobile Discovery
- Run Mobile Inventory
- Deploy Agent
- Upgrade Agent
- Upgrade Scanner
- Run Scanner
- Retrieve Scan File
- Uninstall Agent
- Enrich XML
- Run Rulebase

These actions are described in detail in the [Update Model \(Administrator or IT Manager\)](#) section in [Chapter 8, Using the Device Manager](#) in the *Network Data Analysis Guide*.

- After you specify an action, you can then specify a device group, a profile, or All devices.
- ▶ If the database contains a large number of devices, and you request an update for many (or All) devices, this update may take a long time.

Scanning Devices for Inventory Data

Using Scanners is a powerful way to collect device data. Scanners are distributed to individual devices from the server using the Agent. You can scan each device for its hardware components, and to collect a list of the software installed. The server maintains a schedule dictating which computers should be scanned and when.

Automatic Mode

- ▶ Deploy agents to all computers where you want automatic scan scheduling and scan file retrieval to take place. Refer to the “Setting Agent Deployment Accounts” and the “Setting Up Scanner Schedules” chapters in the *Installation and Initial Setup Guide*.

When a particular computer needs to be scanned, the server contacts the agent on the computer, sends a copy of the appropriate Scanner configuration file and a copy of the latest Scanner to the computer (if the Scanner on the computer needs to be updated), executes it, and retrieves the resulting scan file.

Manual Mode

In Manual Deployment mode the DDM Inventory Server is not used to schedule and launch scans. For example, if the scans will be launched from login scripts or on non-networked machines. The Scanner Generator generates self-contained Scanner executables that consist of a combination of the Scanner executable and configuration file.

If you are doing the inventory manually using manual deployment mode, you do not need the agent.

For both modes, retrieved scan files are stored in a directory on the server. The XML Enricher polls this directory for new scan files and processes them so they can be added to the server's inventory database. It also enriches the scan files with application data and stores these as compressed XML files.

For a complete description of what the XML Enricher does, and how to use it, see the *Configuration and Customization Guide*.

To learn how to generate scanners, also see the *Configuration and Customization Guide*.

Scanning Devices for Application Utilization Data

In addition, you can also enable the agent plug-in that collects software utilization information. The agent software utilization plug-in generates individual utilization files, one per day when it runs up to the maximum period for which utilization data is collected.

In addition, it also produces a summary file for the entire utilization period. This file is an XML data file compressed using gzip (Compressed XML utilization). The XML is encoded using the UTF-8 encoding.

The XML Enricher does the following during its processing:

- Extracts and parses the XML data out of the stored file.
- Calculates the software utilization for each recognized application and adds this information to the enriched scan file.
- Adds a 'Utilized' flag to the file attributes, calculates and adds utilization figures for executables that were executed.

Filter Out of Device Database

DDM Inventory can filter out certain unmanaged devices that you do not want to see on your map or in your DDM Inventory database.

You can change these filter settings in **Administration > System Configuration > Input filters > Devices not to add to the database:**

- Unmanaged devices which are MAC plus IP
 - Unmanaged devices which are MAC plus IP and not pingable
- Unmanaged devices which are MAC-only
 - Unmanaged devices which are MAC-only with unknown OUIs
- Unmanaged devices which are IP-only
- Scanned-only devices
- Virtual-only devices
- Mobile devices

All of these filter settings change which devices can be monitored by DDM Inventory. If a device is filtered so it does not appear on the Network Map or in the database, the device data will be found at **Status > Device Status > Filtered devices.**

Add to Device Database

Once a device has been modeled, the device is added to the Real-time database. This database is hosted on the DDM Inventory server and is managed by an embedded MySQL server. The device information is also replicated in the Reports database, which is managed by the same MySQL server. In general, additional details are attached to the device information at this time.

The Reports database is used to generate most of the reports that can be accessed through the User Interface (UI). It can also be used to populate the asset management database. Using the scenario provided with Connect-It 3.6 or later, transferring the DDM Inventory data to AssetCenter is a simple matter.

The Reports database schema is fully documented (the documentation is available through the UI) and is not user-configurable.

Rulebase

The Rulebase is the component of DDM Inventory that assigns a device type to your device model. The Rulebase is a large database containing information on hardware available for sale by a wide range of manufacturers. When DDM Inventory has a model, it will search the Rulebase to find that particular manufacturer and other pertinent information.

The Rulebase is responsible for giving each device an appropriate icon (as seen throughout the DDM Inventory interface), and making sure the basic device information is up to date.



Rulebase updates are made available on a regular basis. Check the support website for updates.

Data Going into the Rulebase

The following fields are passed to the Rulebase so it can make a determination about the device:

- system.sysDescription
- system.sysObjectID
- DNS name
- MAC address
- NetBIOS name
- forwarding information (whether or not the device is routing IPv4 or IPv6)
- read community string
- write community string
- number of ports

The following fields are available for scanned devices:

- Operating System hwHostOS
- BiosProductName hwBiosMachineModel
- BiosProductManufacturer hwsmbiosSystemManufacturer
- BiosChassis hwsmbiosChassisType
- BiosDescription hwBiosMachineDescription
- Operating System ServicePack hwOSServiceLevel

The following fields are passed to the Rulebase for mobile devices:

- Mobile manufacturer
- Mobile model
- Mobile device type
- Mobile OS

The following fields are passed to the Rulebase for VMware host systems:

- The model of the hardware (for example: ProLiant DL365 G1)
- VMware server type and version (for example: VMware ESX Server 3i 3.5.0)

Creating a Device Type

The DDM Inventory Rulebase then assigns a device type to each device. The device type can include the following characteristics:

- Model
- Model URL
- Model Manufacturer
- Family
- Family URL
- Family Manufacturer
- OS
- OS URL
- OS Manufacturer
- Network Function
- Network Function URL
- Network Function Manufacturer
- Software Historical Manufacturer
- Software Historical Manufacturer URL
- Software Present Manufacturer
- Software Present Manufacturer URL
- Historical Manufacturer
- Historical ManufacturerURL
- Present Manufacturer
- Present Manufacturer URL
- Icon
- Title
- Tag
- Priority
- UNSPSC model
- UNSPSC description
- UNSPSC application
- UNSPSC operating system

Adding Rules to the Rulebase

If a specific rule is not in the Rulebase, Hewlett-Packard will add a rule for you, provided that:

- Your DDM Inventory software is under warranty
- You can identify the devices by model or family (Hewlett-Packard would appreciate the URL for the manufacturer's web site whenever possible)

- You provide Hewlett-Packard with a CSV copy of your inventory containing the device.

Some devices and device families may not match a specific identification rule. Such rules are likely to make good assignments for companies with small product lines and less accurate assignments for companies with large product lines. This is because these rules are based on:

- Advance classification of product lines; that is, some devices belonging to certain product lines can be identified by the beginning of the OUI
- Pre-identification of specific devices; that is, some devices can be identified because the manufacturers make only switches



These rules may make incorrect assignments. You should contact Hewlett-Packard to request additional specific rules for devices if this happens.

For some devices without SNMP management, the Rulebase can apply rules based on information about the MAC address and OUI of the device. For each MAC address, the Rulebase identifies the most probable device class, based mostly on the OUI. (Occasionally, manufacturers assign blocks of MAC addresses to specific products, which allows the Rulebase to make more specific identifications.)

The Rulebase also identifies the probability that each non-SNMP device may actually be an SNMP managed device providing network connectivity (such as a gateway, router, concentrator, or switch). SNMP managed devices can appear not to be managed when the device's IP address has been included in the list of device groups (see **Administration > Discovery Configuration > Device Groups**) or when the community string for the device has not been included in the list of community strings (SNMPv1/v2) and users (SNMPv3) (see **Administration > Discovery Configuration > Configuration Profiles**). In such a case, you should install or enable the SNMP agent for that device. (You may also need to modify the address scope or community strings.)

As with devices with SNMP management, class assignments for devices with no management work well for companies with small product lines and poorly for companies with large, varied products lines. Larger companies sometimes employ the same OUI for different products, but also use different OUIs for one product.

There is also a capacity to assign icons to unmanaged devices based on information contained in the NetBIOS and domain names deleted. For example:

- a device named "PRINTER3RDFLOOR" or "PRT3RDFLOOR" could be assigned a printer icon
- a device named "marysworkstation" could be assigned a workstation icon
- a device named "webserver.example.com" could be assigned a web server icon

Only the initial segment of the domain name is considered.

Domain and NetBIOS name interpretation is low priority. It never takes precedence in a situation where more accurate information is available. The rules used are not case sensitive.

In some cases information from one device's MIB is used to describe another device. For example if two Cisco devices are directly connected, the CDP information from one **get** gives you information (like SysName, SysDescr, etc.) for the second one, even if the second one is not SNMP managed. This could be used to assign icon/device type to an unmanaged device because we have rules based on information collected from "remote" devices.

Printers

Finally, DDM Inventory can identify printers attached to printer servers. Many printer servers (both internal and external) do not provide enough information in their System Description to allow for accurate identification of the specific model of printer attached.

The DDM Inventory Rulebase uses information that may be found elsewhere in the device MIB. For example, the System Description of this Hewlett-Packard printer server contains the following:

- HP ETHERNET MULTI-ENVIRONMENT,ROM H.08.01,JETDIRECT EX,JD34,EEPROM H.08.05

Note that this does not provide any information about the printer. The Enterprise MIB contains additional information that allows the Rulebase to identify the printer server as J3263A and the printer model as a HP LaserJet 5.

Table Reader



The Table Reader applies only to SNMP-managed devices.

Unmanaged MAC-only devices (and MAC-only devices with unknown OUIs) will be discovered after DDM Inventory completes modeling devices with the bridge table reader.

Devices can also be discovered as a MAC+IP pair from information collected from router arp caches.

Connectivity information comes from the Table Reader. If DDM Inventory has identified the device as a bridge, the Table Reader reads its bridge tables. If the device has been identified as a router, or if “Force ARP Table Read” has been enabled in **Administration > Discovery Configuration**, the Table Reader reads its ARP table.

Generate Add Event

As soon as the device is added to the database, a “Device Add” event is generated to alert you that DDM Inventory is now monitoring the device. You can see the event listed in the Events Browser (see the *Network Data Analysis Guide*).

Choose a Scanning Method

Beginning with DDM Inventory version 7.50, you can choose between agent-based scanning and agentless scanning.

What is an Agent?

An Agent is installed as a service on a remote device. This service enables the device to be securely scanned at any given time.

On supported Windows platforms, an administrator can instantly deploy this service on devices using a service like Windows Remote Administration (RPC). Go to **Help > Compatibility Matrix** in the DDM Inventory GUI to see all supported platforms.

On Unix client devices, a compressed tar archive is provided for each platform to allow manual or scripted installation.

Once installed, the Agent is capable of communication with the Agent Communicator Service (ACS) on the DDM Inventory Server.



If an Agent is manually removed from a Windows device, DDM Inventory can detect that and automatically redeploy the Agent.

Agent-Based and Agentless Scanning

If you are using agent-based scanning, the Agent listens and performs those tasks requested by DDM Inventory. For example, it can deploy a Scanner, execute a scan, or transfer a scan file to the Server. In agent-based scanning, if you want to schedule automatic scans, the Agent must be installed on every workstation that will be part of the inventory process. In this scenario, a newly discovered computer cannot be scanned without first installing the Agent. If you want to perform a manual inventory—that is, you plan to manually deploy and execute the scanner—you do not need the Agent on the remote device.

Agentless scanning allows a device to be scanned without deploying an Agent to that device. In this case, DDM Inventory creates a secure connection to the client device, copies a scanner to the device, executes the scanner, and retrieves the scan data. All operations are encrypted and require knowledge of administrator credentials that are valid for the device or group of devices being scanned.

Agent profiles include communication credentials that the DDM Inventory Server uses to either perform agentless scans or automatically deploy the Agent (Windows only). On Windows platforms, it is necessary to configure communication credentials for both agent-based and agentless scanning. For other platforms, it is necessary to configure communication credentials for agentless scanning only.

Securing Agent Communication

For agent-based scanning, the DDM Inventory Agent Communicator Service (ACS) on the DDM Inventory Server communicates with the Agent on the client device. The Server always initiates the communication. The client device where the Agent is installed is never able to request information from the DDM Inventory Server.

In agent-based scanning, communication uses a 100% standard authentication/encryption method. This is identical to the method used to secure web servers. Communication is initiated by the DDM Inventory Server using an HTTPS 2048-bit RSA (Rivest-Shamir-Adleman) public / private key method. RSA is a security method that uses a two-part key: The private key is kept by the owner, and the public key is published. Data is encrypted by using the recipient's public key, which can only be decrypted by the recipient's private key.

For agentless scanning on Windows client systems, DDM Inventory uses the remote management capabilities of the Windows operating system to create the secure connection. Data is encrypted the same way that it is for agent-based scanning using 3DES / RSA 2048 encryption.

For agentless scanning on UNIX, Linux, and Mac OS X client systems, Secure Shell version 2 (SSH-2) is used to create the connection.

Further information

For more information on agent-based and agentless scanning, see the *Installation and Initial Setup Guide*.

For information about the platforms the agents are available for, see the *Compatibility Matrix*.

Schedule Scan

Schedule Scan is a feature that allows you to control when the scanners are run on the workstations.

For more information on creating scan schedules, see the *Installation and Upgrade Guide*.

Automatic Deployment of Scanners

In most cases, once an Agent has been installed on the computer, the Scanners can be automatically deployed as that computer is discovered with DDM Inventory, and executed as needed.

This mechanism makes Scanner deployment an easy task, as opposed to situations where deployment has to rely on network login scripts or manual intervention. Thus, the accuracy and completeness of the collected inventory data can be very high.

Scheduling of Scan Execution

It is possible to specify a schedule for computer scanning. The schedule serves at least two purposes.

- First, although the execution of a scan is designed to be as unobtrusive as possible to the user of a computer, some users do notice and find it distracting. So scans can be scheduled to run at a time of day that tends not to conflict with users.
- Second, the accuracy of the inventory depends on the frequency of scan execution. For example, some users want the data refreshed every week, others every month. So the frequency of scans can be specified.

Collection and Storage of Scan Files

A Scanner writes a scan file to the local disk of the scanned computer, and the scan file is transferred to the server for storage and processing. There are a few ways in which the scan file can be transferred:

- The server contacts the computer and transfers the scan file from the computer. This is the typical case for computers that are permanently connected to the network. The collection of scan files is scheduled and controlled to minimize impact on the network. For example, you can specify what times of day are appropriate for scan file collection, how many files can be transferred in parallel, and how much network bandwidth scan collection is allowed to consume.
 - ▶ Collection of scan files is decoupled from the execution of a scan. You can schedule scans for one time of day, but collect the scan files some time later using a different schedule.
- Some computers, for example laptops, are only occasionally connected to the network. In this case, the scan file can be transferred whenever that computer connects. Since the connection speed may be slow and the connection time short, the transfer mechanism gracefully recovers when the connection is interrupted and can be resumed when the connection is re-established.
- Some computers may never be network accessible to the server, for reasons of network topology or security. In this case, it is the responsibility of the administrator to transfer the scan files from such computers to the server.

Scan File Enrichment

Once a scan file is transferred to the server, it is further processed to recognize software applications (XML enrichment process) and added to the inventory information stored in the Inventory Database. The resulting enriched scan file is stored on the server for subsequent access by tools such as Viewer, Analysis Workbench or Connect-It.

This enriched scan file is always stored in compressed XML format. At most one scan file for each computer is stored, and the name of the scan file is normally derived from the Asset Tag uniquely identifying the machine. The enriched scan files are optionally backed up along with other server data.

Further information

For further information about the XML Enricher, refer to the *Configuration and Customization Guide*.

Poll Device

- ▶ Only SNMP managed devices are polled. Scanned-only devices are not polled.

Once there are device models in the database, the pollers begin collecting data from the devices. There are three pollers: the Realtime Poller, the Resource Poller, and the Environmental Poller.

The Realtime Poller also reads the device's MIB to get information about the device's traffic and connectivity on all of its ports. The resulting data is passed to the Mapper to determine connectivity.

How Long Does it All Take?

Generally, “discovery” can work like this:

- A device will be discovered based on the discovery ping rate and the device groups you have established.
- The time taken to create a device model will vary depending on the type of device, and the number and order of the following items:
 - Community strings for SNMPv1 and SNMPv2
 - Users for SNMPv3
 - Credentials for mobile or virtual devices

The time may increase for some devices if DDM Inventory needs to try several community strings (or users for SNMPv3) to access the device MIB—or credentials to access the mobile or virtual host server.

DDM Inventory also uses scan files to supplement a device model, and that process is quite different. Once the device has a model in the database, an Agent can be deployed to that device. After the Agent is deployed and the DDM Inventory server can contact it, a Scanner is deployed. The time may increase depending on how many Agent Deployment Accounts are configured to access the devices.

There are many factors that affect the time it takes to scan a particular device. The scanner configuration is a major factor. Hardware-only inventories take from a few seconds to a couple of minutes to complete depending on the configured hardware tests and selected scanner priority. On the other hand, the software scan usually takes much longer and depends on the chosen configuration, the scanner priority, and the number of files and directories available on the computer to be scanned. Other factors include the speed of the computer and its hard drive. As a rough estimate, a scan of an average workstation using the default scanner (hardware and targeted software) takes around ten minutes.

The DDM Inventory server is able to communicate with up to 76 agents at the same time. Server to agent communication includes uploading scanners and their configuration, starting scanners, and collecting scan file results. However, these communications are usually brief, so the server is able to launch many more than 76 simultaneous scans at the same time. The overall time it takes to scan a network depends on how long it takes to scan each computer and on other factors, such as available bandwidth. It is recommended that you perform a pilot inventory that scans a subset of all managed devices in a particular environment using configured settings to get a better idea of the exact times required.

DDM Inventory is Always Discovering

This process runs the entire time DDM Inventory is in operation.

Also, every device is re-modeled at the device remodeling interval specified in **Administration > Discovery Configuration**.

This way, DDM Inventory constantly strives to present you with an updated view of your network, and constantly strives to improve the accuracy and depth of that view.

DDM Inventory's Topology Map

To calculate network connectivity, DDM Inventory uses a probability engine with a variety of patented algorithms.

On each device, DDM Inventory considers the following:

- the source address capture information
- link training
- bridge tables
- vendors' proprietary tables (including Cisco's CDP)
- VLAN bridge tables

Bridge tables are huge and DDM Inventory needs a lot of computation power to analyze them. DDM Inventory will delete most of the information in the raw bridge tables and retain the information needed to build a virtual bridge table.

At this point, DDM Inventory looks at all the bridge tables and figures out which ports are up ports (one network device with bridge tables connected to another network device with bridge tables), and which ports are down ports (network devices with bridge tables connected to other devices without bridge tables).

Once the ports have been classified as up or down, the bridge tables have been reduced so the computing required for determining connectivity gets much simpler. Typically DDM Inventory deletes 95% or more of the information in a bridge table. From this, DDM Inventory generates port-to-port connections.

DDM Inventory also uses traffic patterns to determine connectivity. DDM Inventory is self-similar or fractal, so all the traffic patterns are different.

Traffic pattern matching works by matching the traffic going in and out of one interface, against the traffic going in and out of the other interface. This data is automatically correlated. Based on probability DDM Inventory will determine connectivity between two devices once a threshold is met, and then say that two ports are connected based on traffic information. DDM Inventory also uses sorting techniques to reduce this intractable N^2 problem down to a usable $N \cdot \log(N)$ computation. Additional heuristics and table logic are introduced to clean the data and optimize the success of resolving connectivity.

How Much Network Bandwidth Does the Server Need?

There are many factors that contribute to the network traffic caused by DDM Inventory. The best practice is to connect the server to a major backbone switch. It is estimated that the traffic would total 3-4% on the 10MB dedicated link between the server and the switch. On a 100MB or 1GB link, the impact is proportionally smaller.

The traffic initiated from the server is heaviest on that link to the switch. From the switch, the traffic going to the network is dispersed. It is impossible to say exactly how much bandwidth will be taken by DDM Inventory, but in this section, you can read about some of the influences you may want to consider.

DDM Inventory must contend with many different types of devices, each of which will contain varying amounts of data to be collected. Also, the many settings in DDM Inventory can change how often the data is collected. This makes it difficult to offer an idea of how much bandwidth will be needed in any particular situation.

Also, on the Device Manager, check some of the Statistics graphs such as:

- SNMP Bytes
- SNMP Frames
- ICMP Frames

Discovery Ping Rate

The Discovery Ping Rate (**Administration > System Configuration > Network devices**) is one source of traffic. The ping sweep occurs in the background to look for new devices that have not been found previously. If you turn the ping rate down, it will take longer to discover new devices in your network.

If you turn DDM Inventory's Ping feature off (**Administration > System Configuration > Discovery services**), new devices will not be found through this method, the active part of the discovery will not generate any traffic on your network.

If you have configured DDM Inventory to ping a large IPv4 range containing very few devices, there may be some network impact as pinging non-existent IP addresses will cause ARP broadcast requests.

DDM Inventory has been configured to limit the ARP broadcasts it generates. However, DDM Inventory may ping devices on the far side of a router. You should check your router configuration if broadcast levels become unacceptable for your network. On your router, consider the following:

- Increase ARP cache size
- Increase ARP aging time
- Reduce ARP retry rate

Table Reading and Polling

Table reading and polling produce the majority of network traffic from the DDM Inventory server. These functions provide:

- Connectivity information
- Discovery of devices (for example, MAC-only devices)
- Collection of statistics to help with finding the topology

A poll is really one frame out and one frame back in most cases. The number of polls for a device will depend on the number of ports in the device, and the number of attributes collected for each port (for example, collisions, broadcasts, etc.). The device itself is also pinged in each poll cycle.

Consider the fact that collecting statistics on a router with 200 ports requires a lot more effort than collecting statistics from a workstation with one port.

Scanning/Agent Deployment

The following two sections apply only to agent-based scanning. For information about agentless scanning, refer to [Two Types of Scanning: Agent-Based and Agentless](#) in the *Installation and Initial Setup Guide*.

Initial Agent Deployment

Initial automatic agent deployment is available for Windows NT/200x/XP. The Windows agent installation is around 1MB in size (the exact size can be seen by looking at the size of the agent .msi file located in the LiveAgents subdirectory of the DDM Inventory data directory) – it is transferred to each computer where the deployment is taking place. The server can run up to 50 simultaneous agent deployment sessions at any one time. The exact number used (default is 25) is specified **Administration > System Configuration > Agent communication > Agent deployment concurrent sessions**.

Custom automatic deployment can be done for other platforms by writing a custom deployment script. For example, agent installation on UNIX may be done via SSH or RSH, etc. The exact bandwidth used by this method depends on the detailed implementation.

Agent Upgrades / Scanner Deployment / Scanning

When performing any of these activities, by default the DDM Inventory server can communicate with a maximum of 80 agents at any one time. This maximum number can be configured in **Administration > System Configuration > Agent communication > Agent deployment concurrent sessions**.

- **Agent upgrade:** when doing an agent upgrade the agent media (an .msi file for Windows and .tar.z file for UNIX) are transferred to the remote computer. The size of these files depends on the platform and can be seen by looking at the live agent media directory (the LiveAgents subdirectory of the DDM Inventory data directory).
- **Scanner Deployment:** when the scanner has not been deployed yet or an old version of the scanner is available on the remote computer, the new copy of the scanner is deployed. The scanner size varies depending on the platform – the exact sizes can be seen by looking at the Scanners subdirectory of the DDM Inventory data directory.
- **Scanning:** unlike agent and scanner deployments/upgrades, which are usually one-off events, the scanning activity involves the DDM Inventory server asking the agent to run the scanner regularly according to a specified schedule. After the scanner has finished executing, the scan file is saved locally and then transferred to the server. The size of the scan file varies depending on the size of the box and how the scanner is configured.

The network bandwidth used for scanner deployment/upgrade and the scan file retrieval can be capped for each DDM Inventory server to agent connection—this is specified in the Agent configuration profile.

Benchmarking Data

The DDM Inventory team has conducted tests to benchmark statistics about network traffic caused by the DDM Inventory server.

This test was conducted in a lab with eighty machines. The test began with 20 machines attached to the network and available for discovery. Then, 24 hours later, 20 additional machines were added. Finally, after 24 more hours, 40 additional machines were added, for a total of 80 machines attached to the network.

This incremental approach allowed the test team to gather statistics about the increase in network traffic vs. the number of machines on the network.

Description of the Two Test Series

Two series of tests were completed.

Test series 1 captured packets when DDM Inventory was set to simply discover devices in the network. All devices had SNMP enabled.

Test series 2 enabled the deployment of DDM Inventory agents and scanner execution, along with the network discovery capabilities.

The benchmark tests were considered successful when:

- SNMP enabled on all of the devices
- A full Discovery and deployment occurred
- Scans were created and retrieved.

Reporting Methodology

The reports consist of metrics from the Ethereal 0.10.0.3 network sniffer utility. Data from the Ethereal utility was then exported to Microsoft Excel, and calculations were made on that data.

Network Test Results

When the number of devices on the network doubled (from 20 to 40) with the second test iteration, the network traffic statistics did not double; the overall number of bytes transferred increased by only 20%. This shows that DDM Inventory is scalable in a way that does not proportionally impact the volume of network traffic.

The average packet size almost doubled when the number of devices was increased in the third iteration (from 20 to 80 devices).

Table 1 Network Test Series 1

Statistic	Value with 20 devices on the network	Value with 40 devices on the network	Value with 80 devices on the network
Between first and last packet	86,412.572 sec	86,415.326 sec	86,414.411 sec
Packets Captured	436,445	478,244	310,000
Avg. Packets/sec	5.051	5.534	3.587
Avg. Packet size	68.000 bytes	78.000 bytes	137.000 bytes
Bytes	29,885,826	37,573,212	42,654,214
Avg. bytes/sec	345.850	434.798	493.601
Avg. Mbit/sec	0.003	0.003	0.004

Scanner Test Results

When scan files are added, the number of bytes transferred on the network grows greatly. There is almost a 500% growth in total bytes compared to the test that did not include scan files. The average packet size also grew close to 500% as compared to the non-packet test.

Comparing the results from this test alone, the growth in network traffic and packet size does not grow in proportion to number of devices on the network.

The growth when the number of devices doubled on the networks was about 30%. Packet size growth was close to 10% when the number of devices doubled.

Table 2 Network Test Series 2

Statistic	Value with 20 devices with scanner deployment	Value with 40 devices with scanner deployment	Value with 80 devices with scanner deployment
Between first and last packet	86,415.804 sec	86,413.504 sec	86,416.253 sec
Packets Captured	770,696	938,701	998,428
Avg. Packets/sec	8.918	11.295	12.052
Avg. Packet size	171.000 bytes	195.000 bytes	221.000 bytes
Bytes	132,384,807	183,085,794	221,580,176
Avg. bytes/sec	1531.951	2202.967	2674.692
Avg. Mbit/sec	0.012	0.018	0.021

WAN

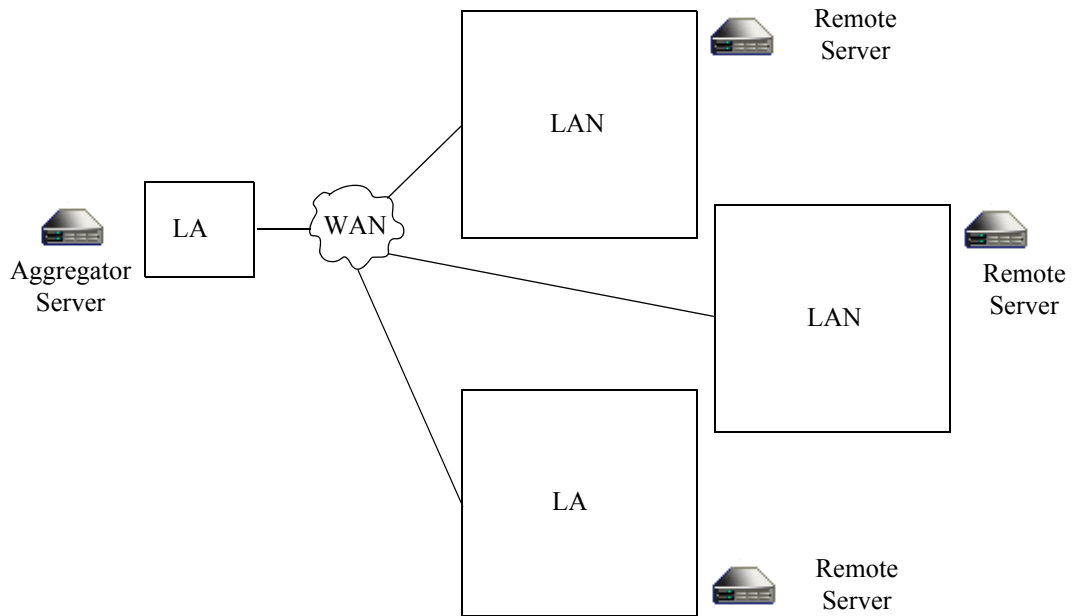
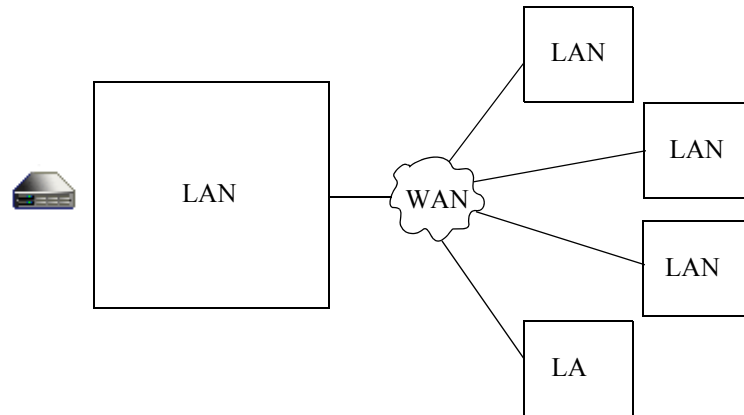
The bandwidth on a WAN link depends on the number of devices in your device groups, the types of devices, and the many settings available in DDM Inventory.

However, the real impact depends on the amount of bandwidth available, how much is used by normal network traffic, and what levels of extra traffic you are willing to accept.

Over a slow link, it would not be practical to fully manage a large network. Depending on the various parameters, you may only want to monitor the core devices. If you need to manage your entire network, it may be wise to get a second DDM Inventory server on the other side of your WAN link. The two servers could be aggregated and share data.

There are many other considerations in a WAN scenario. In a large network (for example, 10,000 devices) with high rates, assume that the DDM Inventory server will use 5% of a 10MB link for network management.

Using DDM Inventory in a WAN:



Aggregating DDM Inventory servers

What Is an Aggregator?

The Aggregator is a DDM Inventory server with a license that also allows it to collect and combine data from several DDM Inventory servers in your network. The health data is combined into one Aggregate Health Panel, so you can see the status of the entire network. An Aggregator also allows you to access other individual DDM Inventory servers without logging into them directly.

You can aggregate up to 50 DDM Inventory servers with a maximum of 500,000 devices. However, the more servers you aggregate, the more slowly the Aggregator processes the data. While performing as an Aggregator, the DDM Inventory server can also serve as a regular server, monitoring up to 100 devices.

It is important to remember what is aggregated, and what is not. The following functions are aggregated:

- Health Panel
- Alarms Viewer
- Events Browser

Also, with an Aggregator, you have an integrated data source for exporting onto data access applications using the Open Database Connectivity Standard (ODBC). See the *Network Data Analysis Guide* for more information.



If a remote server is not available, the Aggregator uses the last available imported Health Panel for that remote server. You can detect the status of a remote aggregator server by navigating to **Aggregate Status > Aggregate server health**.

How Do I Use the Aggregator?

An Aggregator DDM Inventory server works like a regular DDM Inventory server. The Aggregator has an extra license that allows it to collect data from the other DDM Inventory servers in your network. The Aggregator can also be responsible for monitoring a specific part of the network, while simultaneously collecting data from other DDM Inventory servers and presenting them in the Aggregate Health Panel.

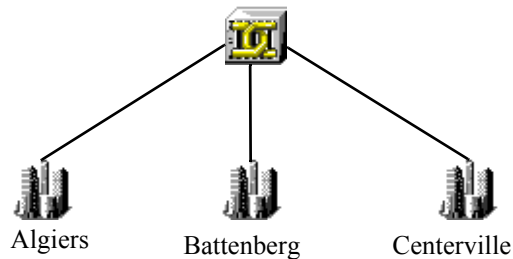
There are many ways you could set up aggregation in your network, depending on the network topology and how many DDM Inventory servers you have installed.

- You can use the Aggregator as a regular server to monitor a part of your network, as you would with any of your DDM Inventory servers.
- You can use the Aggregator as a regular server to monitor only the backbone of your network, your important routers and servers, as well as the other DDM Inventory servers.

If you have the resources available, we recommend the second option. You can use the other servers to monitor the subnets, but this will give you a real focal point from which you can access the rest of your network.

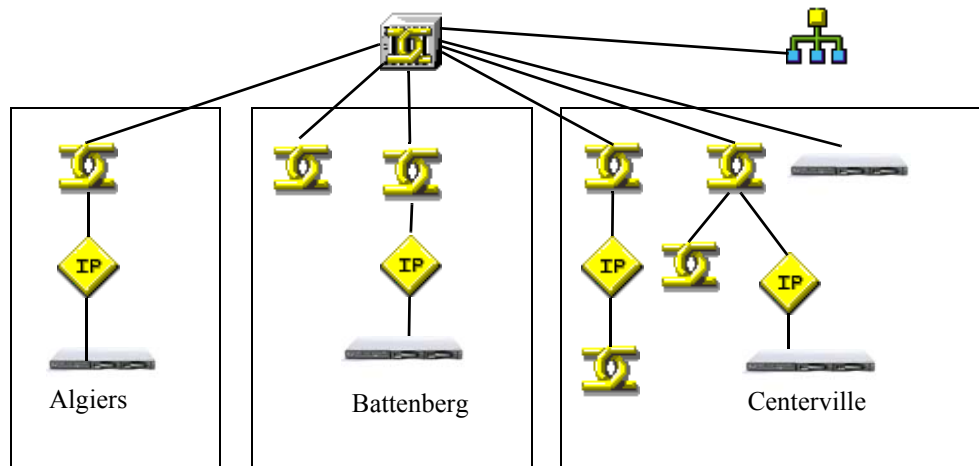
Setup Example

Suppose that you work with a business, ExampleCorp, that has offices in three cities: Algiers, Battenberg, and Centerville. Each office has 6,000 devices in its subnetwork.



Ideally, you would have purchased 4 DDM Inventory servers: one for each office, and one to act as an Aggregator for the central office (in Centerville).

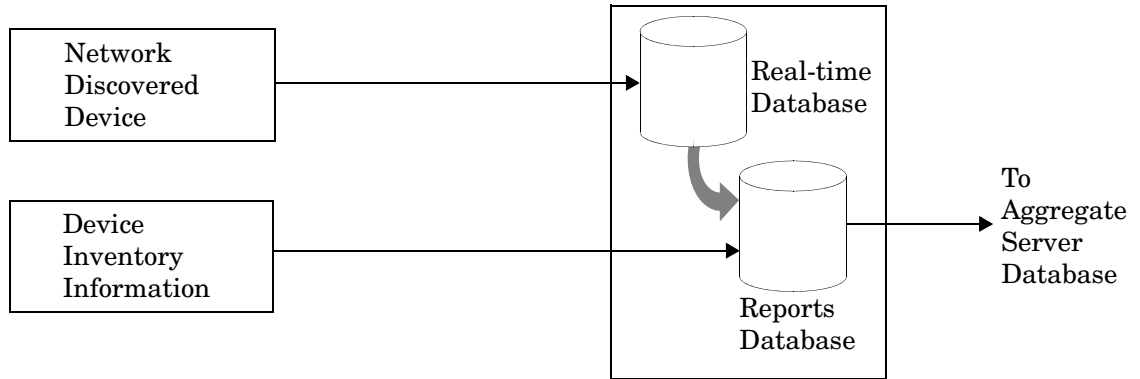
If you set up the Aggregator ranges to include only DDM Inventory servers and routers, the resulting network may look like this:



Aggregator Data Transfer

In order to understand what information is made available from the remote DDM Inventory servers to the Aggregate server, you will need to understand the configuration and data transfers that take place on the remote servers.

The following diagram illustrates the internal data transfer for each remote server configured before the data is sent to the Aggregate database server.



For performance reasons, the DDM Inventory database is actually implemented as two separate databases on each remote server:

- The Real-time database
- The Reports database

The Real-time database contains very basic information gathered about each device during the Discovery process. The Reports database contains this basic information plus any detailed Inventory information that is collected for each device.

By keeping the two databases separate, the Discovery process can run at full speed without being negatively impacted by either the Inventory process or any report generation activities that might be initiated from the UI.

The information collected by the Inventory process is device-specific, and it is added to the device's model information in the Reports database. It is merged with information that was collected about that device during the Discovery process.

Data Transfer from Real-Time Database to Reports Database

There are five categories of data periodically transferred from the Real-time database to the Reports database: events, hourly summary, hardware discovery data, reports, and inventory data. For each category, data is transferred in two phases:

- **Extract:** The relevant data is extracted from the Real-time database and stored in tab-separated value (TSV) files.
- **Import:** At the scheduled time, DDM Inventory imports the data from the TSV files into the Reports database.

Each category of data has its own data transfer schedule. The following five sections indicate the default schedule for the extract and import phases and the Reports database tables that are updated for each data category.

Events

The amount of data and the frequency with which it is imported into the database is configured in:

Administration > System Configuration > Attribute export schedule

- The default frequency is 5 minutes. By default, data is only imported into the database when attributes are in an alarmed state. **Extract:** Every 5 minutes starting on the hour.
- **Import:** Every 5 minutes starting on the hour.

- **Tables Updated:** Attribute, AttributeState, ConnectionEvent, Event

Hourly Summary

This information is used to report statistical information on the devices in the network.

- **Extract:** Every hour.
- **Import:** Every hour.
- **Tables Updated:** HourlySummary

Hardware Discovery Data

Hardware Discovery Data can be imported into the database on demand from:

Administration > Data Management > Update reports database

This action performs both the extract and import of the data. However, the normal automated schedule for performing those tasks is as follows:

- **Extract:** Daily at 6 AM. Two copies of TSV files are produced: one for local consumption and one for the Aggregator server consumption.
- **Import:** Daily at 8 AM.
- **Tables Updated:** Device, hwAssetData, IPv4, MAC, Port, SerialNumber, SubComponent, Vlan, VlanObject, Host, VirtualDevice

Reports

DDM Inventory provides numerous reports to help you analyze and understand what your network contains. Used for historical reasons, this information is then updated to the Reports database and associated with each Modeled device.

- **Extract:** Every hour on the hour
- **Import:** Every hour on the hour.
- **Tables Updated:** EventSummary, ReportState, ReportStateSummary

Inventory Data

Inventory information is collected in four different ways depending on the type of device:

- SNMP requests
- Scanners and the XML Enricher
- Virtual device server queries
- Mobile device server queries

Inventory information is stored in the Reports database as it is collected. Specific pieces of inventory information are also stored in the Real-time database for device-modeling and merging purposes.

For example, when the XML Enricher processes a scan file, certain scan file data is passed to the Real-time database. This allows the device models to be updated with data such as IP addresses, MAC addresses, and ports. In addition, the XML Enricher creates TSV files to be imported into the Reports database.

The extract and import schedules depend on the method used to collect the inventory information. For data generated using SNMP requests, the schedule for extract and import is the same as for hardware discovery data. The same applies to virtual device data, which refreshes the additional VirtualDevice and Host tables.

For data generated using the scanners and XML Enricher, the schedule is as follows.

- **Extract:** Once an hour, or after 100 scan files have been processed, the TSV files are transferred to the Reports database import directory.
- **Import:** Every 30 minutes, starting on the hour.
- **Tables Updated:** Application, Company, DailyUsage, Directory, File, FileDirectoryVersion, FileInstance, Release, SoftwareUtilization, SWSSubComponent, User, Version, and all tables whose names begin with “hw” except hwAssetData

For Mobile device data, the schedule is as follows:

- **Extract:** Once an hour, or after 100 mobile devices have been processed, the TSV files are transferred to the Reports database import directory.
- **Import:** Every hour.
- **Tables Updated:** MobileDevice

Data Transfer from a Remote Server to the Aggregator Server

The previous section discusses how data is transferred internally within a Remote server. This section covers how data is transferred from a Remote server to the Aggregate Server.

For example, for hardware discovery data (see [Hardware Discovery Data](#) on page 35), two copies of the TSV files are constructed on the Remote server. One copy of the extracted data is always consumed by the local Reports database on the Remote server, regardless of whether this server is aggregated or not. If aggregation is configured, the second copy is transferred to the Aggregator server to be consumed by the Reports database import process on that server.

The two copies of the TSV files are produced to reduce processing delays: the Aggregator does not have to wait for the local import to be completed on the Remote server before being able to perform its own import.

Data Types Aggregated

Four types of data are aggregated: events, hardware discovery data, reports, and inventory data. These are the same data types that are imported into the Reports database on a single Remote server—except for the Hourly Summary data and Unrecognized File inventory data, which are not aggregated.

During aggregation, data is imported into the Reports database on the Aggregator server. The aggregate import schedules may be different than the local import schedules for any given Remote server. Identical sets of TSV files are used for the local and aggregate import processes.

Aggregated data is imported in two phases:

- **Transfer:** Data is copied from the Remote server onto the Aggregator server.
- **Import:** At the scheduled time, DDM Inventory imports the data into the Reports database on the Aggregator server.

Events

As a device's events change during its Discovery and Inventory life cycle, this information needs to be in the central database (Aggregator) for administration.

- **Transfer:** Transfer schedules can be configured in:
Aggregate Administration > Remote server administration > Remote Server Properties

The default transfer schedule is every 5 minutes starting on the hour.

- **Import:** Every 5 minutes starting on the hour.
- **Tables Updated:** Same as local import.

Hardware Discovery Data

This information is used to report information about the devices found in the network.

- **Transfer:** Daily at 7:00 AM.
- **Import:** Daily at 8:00 AM.
- **Tables Updated:** Same as local import.

Hardware discovery data can also be imported on demand from:

Aggregate View > Aggregate Status > Aggregate server health

Reports

The Reports data provides a summary of the events that have occurred on the remote DDM Inventory servers.

- **Transfer:** Every hour on the hour.
- **Import:** Every hour on the hour.
- **Tables Updated:** Same as local import.

Inventory Data

Again, inventory information is collected in four different ways depending on the type of device:

- SNMP requests
- Scanners and the XML Enricher
- Virtual device server queries
- Mobile device server queries

As previously discussed, the data collected via SNMP request or on Virtual device servers follows the same schedules as the hardware discovery data.

For data generated using the scanners & XML Enricher, the schedule is as follows:

- **Transfer:** The default is to never transfer this data. This can however be overridden in:
Aggregate Administration > Remote server administration > Remote Server Properties
- **Import:** Every 30 minutes, starting on the hour.
- **Tables Updated:** Same as local import except that the Directory, File, FileDirectoryVersion, and FileInstance tables are not aggregated.

For Mobile device data, the schedule is as follows:

- **Transfer:** Every hour.
- **Import:** Every hour.
- **Tables Updated:** Same as local import.

Connecting the Reports Database on the Aggregator Server

For both the Remote server and the Aggregate server the settings used to connect to the Reports database are the same.

The following parameters are needed to connect to the Reports database on the Aggregator server.

Port: 8108

Database name: Aggregate

Username/password: As defined in DDM Inventory.

The Optimum Time to Run a Connect-It Scenario

Connecting to a Single DDM Inventory Server

For an up-to-date inventory, the Connect-It scenario should be run after the Inventory import at 8am.

Connecting to an Aggregator server

For an up-to-date inventory, the Connect-It scenario should be run after the Inventory import at 8am.

RFCs Supported by DDM Inventory

Table 3 Supported RFCs

RFC number	Name
RFC 1155	Structure and Identification of Management Information for TCP/IP-based Internets
RFC 1157	A Simple Network Management Protocol (SNMP)
RFC 1213	<i>see RFC 2011, RFC 2012, RFC 2013</i>
RFC 1285	FDDI MIB (SMT 6.2); <i>see also RFC 1512</i>
RFC 1315	<i>see RFC 2115</i>
RFC 1354	<i>see RFC 2096</i>
RFC 1398	<i>see RFC 1643</i>
RFC 1406	Definitions of Managed Objects for the DS1 and E1 Interface Types
RFC 1407	Definitions of Managed Objects for the DS3/E3 Interface Type
RFC 1493	Definitions of Managed Objects for Bridges (Bridge MIB)

Table 3 Supported RFCs

RFC number	Name
RFC 1512	FDDI MIB (SMT 7.3)
RFC 1513	Token Ring Extensions to the Remote Network Monitoring MIB
RFC 1514	Host Resources MIB
RFC 1516	Definitions of Managed Objects for IEEE 802.3 Repeater Devices
RFC 1643	Definitions of Managed Objects for the Ethernet-Like Interface Types (Ethernet Interface MIB)
RFC 1695	Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2 (ATM MIB)
—	ATM Forum 3.1 UNI specification
RFC 1748	IEEE 802.5 MIB using SMIv2
RFC 1759	Printer MIB
RFC 2011	SNMPv2 Management Information Base for the Internet Protocol using SMIv2
RFC 2012	SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2
RFC 2013	SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2
RFC 2020	Definitions of Managed Objects for IEEE 802.12 Interfaces (100VG AnyLAN MIB)
RFC 2096	IP Forwarding Table MIB (Router MIB)
RFC 2115	Management Information Base for Frame Relay DTEs Using SMIv2 (Frame Relay MIB)
RFC 2233	Interfaces Group MIB using SMIv2
RFC 2576	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
RFC 2668	Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)
RFC 2674	Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions
RFC 2737	Entity MIB Version 2
RFC 3410	Introduction and Applicability Statements for Internet Standard Management Framework
RFC 3411	An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks
RFC 3412	Message Processing and Dispatching

Table 3 Supported RFCs

RFC number	Name
RFC 3413	SNMP Applications
RFC 3414	User-based Security Model
RFC 3415	View-based Access Control Model

Communication Models

Frame Relay

DDM Inventory supports frame relay devices that conform to:

- RFC 2115, which supersedes RFC 1315

Each physical frame relay port may have one or more circuits associated with it. For some devices, DDM Inventory is able to identify the circuits related to each physical port and gather traffic statistics both for the physical port and for each circuit. DDM Inventory can also make connections between devices connected by these frame relay circuits.

The Device Manager Ports panel presents the ports so as to make apparent the association between a physical port and its circuits. For devices on which DDM Inventory is able to do a physical port mapping, each port is displayed in the form *x.y.z*, where *x* represents the slot or card number on which the port *y* is located, and *z* represents the frame relay circuit.

Using a Cisco 7200 router as an example, here's how DDM Inventory arranges the port structure:

```

...
1.5          —
1.6          —
1.7          frame relay physical port
1.7.27      frame relay circuit
1.7.32      frame relay circuit
2.1          —
2.2          —
...

```

If a device supports frame relay but DDM Inventory is not able to map the exact physical ports, each port is displayed in form *x.y*, where *x* represents the MIB-II object *ifIndex* and *y* represents to frame relay circuit. Using a Cisco 2500 router as an example:

```

1          —
2          —
3          —

```


4	frame relay physical port
4.75	frame relay circuit
4.76	frame relay circuit
4.78	frame relay circuit
5	—
6	frame real physical port
6.21	frame relay circuit
6.27	frame relay circuit

The line speed is set for each frame relay circuit. Each circuit should report a Committed Information Rate (CIR).

The CIR has meaning only for frame relay lines. It is used in service-level agreements and contracts for supply of communications bandwidth over frame relay lines. CIR has no functional impact on the performance of frame relay devices. For DDM Inventory to read the CIR from the device, it must have been entered into the device's MIB. If DDM Inventory cannot find the CIR in the MIB, it sets the frame relay circuit CIR to the line speed for that frame relay physical port.

If DDM Inventory has determined the CIR incorrectly, you can use the Port Manager's Port Properties button to redefine it. You may change the interface rate at either end or at both ends.

The following examples and rules describe the effect of setting the interface rate to set the CIR.

Suppose a frame relay line connects device A port 1 and device B port 2. The CIR (A1-B2) is defined from A1 to B2. The CIR (B2-A1) is defined from B2 to A1 and can have a different value from the CIR (A1-B2).

This table shows an example of the effects of setting the CIR.

A1		B2		CIR A1 to B2	CIR B2 to A1
line speed (kb/sec.)	set by user	line speed (kb/sec.)	set by user	line speed (kb/sec.)	line speed (kb/sec.)
100	no	200	no	100	100
100	no	50	no	50	50
100	no	100	no	100	100
100	yes	50	no	100	50
100	yes	200	no	100	100
100	yes	50	yes	100	50
100	yes	200	yes	100	200

The rules that constructed this table are:

- The line speed is read from the device's MIB unless overridden by the user setting it.
- If the line speed is set by the user at one end, the CIR from this end is defined as that line speed.
- If the line speed is not set by the user at an end, the lower speed at either end defines the CIR for an end.

FDDI

DDM Inventory has limited support for FDDI:

- support for the SMT v6.2 MIB (specified by RFC 1285)
- support for the SMT v7.3 MIB (specified by RFC 1512)

DDM Inventory makes FDDI connections based on the MAC address and MIB variables for each device, not based on the FDDI port.

SMT (Station Management) is an integral part of any FDDI implementation. SMT v6.2 can determine the upstream neighbor for an object. SMT v7.3 can determine both the upstream and downstream neighbors for an object.



If you have a device that supports only SMT v6.2, check with the vendor or manufacturer for SMT v7.3 support. This will improve your FDDI connectivity.

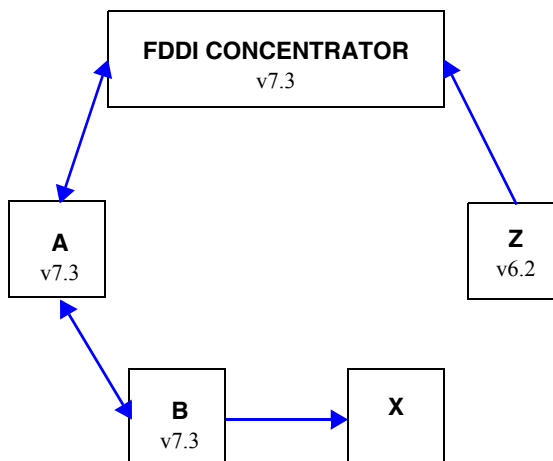
DDM Inventory uses the SMT instance—not the FDDI ports—when mapping FDDI objects. For example, if you have an FDDI concentrator with 8 ports, there is a single SMT instance, so DDM Inventory shows only one uplink port and one downlink port for that concentrator.

If DDM Inventory cannot always close the logical ring for your network, it is because:

- all your FDDI objects have no SNMP management
- all your FDDI objects have SNMP management but support SMT implementations other than v7.3 or v6.2
- at any point in the ring, you have an FDDI object with no SNMP management immediately downstream of an object that supports only SMT v6.2.

To understand this last case, you must realize that the object with no SNMP management (X) is providing no “ring information” about itself to the FDDI ring.

This diagram shows a FDDI ring that cannot be closed.



The only way for the ring to remain unbroken is for the next object upstream (Z) to be able to look back downstream and ask object X about itself. If Z supports only SMT v6.2, then Z cannot see downstream, and therefore the ring cannot be closed.

If DDM Inventory is never able to close the ring, check for objects with no SNMP management followed by an object with support for SMT v6.2 only. This is the only likely cause of a broken ring that will not be immediately obvious.

HSRP

Hot Standby Router Protocol (HSRP) is specifically for Cisco routers. There are other, similar protocols, like the Virtual Router Redundancy Protocol (VRRP) that work with other products. This section is dedicated to HSRP, but DDM Inventory works similarly with VRRP. For more information on how DDM Inventory handles these protocols, contact Customer Support.

HSRP is a routing protocol that allows multiple routers to act as a single “virtual router.” If the main router fails, there are other routers available in “hot standby” mode that will immediately take over the traffic, ensuring constant network connectivity.



For more detailed information on these routing protocols, contact the product manufacturer.

The basic HSRP configuration would be to have a single virtual IP (a.b.c.1) and a set of routers that will respond to this virtual IP (a.b.c.2, a.b.c.3, a.b.c.4, etc.). Only one active router will respond to the virtual IP at any given time.

There are special virtual MAC addresses that are reserved for use with HSRP, in the form of 0000C07ACxx. The same virtual HSRP MAC address can appear in any of the routers (main or standby) that are responding to the virtual IP address.



The routers will all have their own individual MAC addresses as well (appearing in the device MIB).

DDM Inventory may see any combination of IP and MAC addresses for the HSRP routers. It could see the real and virtual MAC for each router, depending on how the routers have been configured.

Since all the routers should respond to DDM Inventory pings and SNMP queries, each physical router (active and standby) should appear on the Network Map. The device model for each router should include its real MAC address.

The virtual IP address may appear on the Network Map if the virtual IP has been seen in an ARP cache and if the routers are SNMP-managed. If the virtual IP does not appear on the Network Map, there will be an unmanaged IP+MAC device, likely connected to the active physical router by a diamond-shaped IP connector device icon.

If the active router is SNMP-managed, the virtual IP will not appear on the Network Map.



If DDM Inventory finds the virtual IP in an ARP cache before it finds the active router, the virtual IP will appear on the Network Map until it is merged with the active router. Once the active router appears on the Network Map, the virtual IP will no longer appear on the Network Map.

Scheduled Events

The majority of data that DDM Inventory uses is constantly being collected. However, some information is collected at a set time every day, while other information is summarized once a day.

This is a list of major events, not a complete list.

Table 4

Time	System event
0005–1900 ^a	summarize statistics for each attribute update Prime configuration summarize events for reports compile and calculate reports
0005–2330 ^b	check devices for deactivating and purging update Health Panel reports (Exceptions, Last Seen, Adds, Deletes, Moves, Changes)

- a. If this series of events is not successfully completed, it will restart in 30 minutes and attempt to complete only the unsuccessful events from the series.
- b. This series only begins once the previous series has finished.

Software Library

The Software Library used in the Application Recognition process consists of a set of Software Application (SAI) files that contain all of the information necessary to deduce the existence of applications from files, registry keys, etc. on a machine. The library also contains license relationship information that allows DDM Inventory to automatically discover actual license requirements even for complex suite-based applications like Microsoft Office, Oracle database servers, etc.

The software library contains information about applications from more than 1000 publishers and covers Windows applications in English as well as some applications in French and German. For UNIX, libraries for HP-UX, AIX and Solaris are included with DDM Inventory.

In addition to the standard libraries, DDM Inventory includes several tools that allow you to create your own library extensions in the form of one or more User SAI files that can easily be applied to the automatic Application Recognition process.

Knowledge Updates

The Rulebase, Software Library and various other data items are updated by Hewlett-Packard on an ongoing basis and updates are made available regularly as DDM Inventory Knowledge updates.

A knowledge update consists of a securely signed archive containing all of the latest data files necessary for your DDM Inventory server. Once downloaded and copied to the appropriate directory on the server, the knowledge update is automatically verified as authentic and applied to the system. Refer to the “Installing Knowledge Updates” chapter in the *Installation and Upgrade Guide* for further information.

3 Terms and Concepts

This chapter is divided into two basic categories:

- [Network Terms and Concepts](#) on page 46 (to review terms and concepts common to network management)
- [DDM Inventory Terms and Concepts](#) on page 52 (to learn terms and concepts unique to this product)

Network Terms and Concepts

These terms and concepts are common to networks and network management. They are not unique to DDM Inventory.

SNMP

Defined by the Internet Engineering Task Force (IETF) in RFC 1157, Simple Network Management Protocol (SNMP) is the protocol that governs network management, and network device monitoring.

MIB

Management Information Base. This database of network management information is used by SNMP. The information contained in this database helps define each device by giving specific information about the device and manufacturer.

Domain Names

Syntax: <localhostname>.<domain>

Example:supt75.SUPPORT-PDC

On the Internet, a hostname is a domain name assigned to a host computer. This is usually a combination of the host's local name with its parent's domain name. A hostname is the unique name by which a network-attached device is known on a network. The hostname is used to identify a particular host in various forms of electronic communication such as the World Wide Web. A domain name allows you to refer to the DDM Inventory Server by a name rather than an IP address. The hostname is translated into an IP address by the local hosts file or the Domain Name System (DNS) resolver.

Address Types

The two main types of numeric address are the IP address and the MAC address.

IP Address

An IP address was intended to be a unique number identifying a unique device or port of a device.

When you see the term “IP address” with no qualifiers in DDM Inventory, it means that either an IPv4 address or an IPv6 address is acceptable. The 32-bit address space of IPv4 addresses puts severe limits on the number of unique addresses available, and the supply is fast running out. The IPv6 128-bit address space was created to address this problem.

IPv4 Address

An IPv4 address contains four sections separated by periods (or “dots”). Each section, called an octet, contains 8 bits expressed in decimal (0–255).

Example: 192.168.96.1

IPv6 Address

An IPv6 address contains eight sections separated by colons. Each section contains 16 bits expressed in hexadecimal (0000–FFFF).

Example: 1234:5678:9ABC:DEF0:1234:5678:9ABC:DEF0

To make it easier to remember and type an IPv6 address, you can use a double colon (::) to indicate multiple contiguous sections of zeros. You can also omit leading zeroes. For example, you can simplify address 0123:0000:0000:0000:0004:0056:789A:BCDE to 123::4:56:789A:BCDE.

MAC Address

A MAC (Media Access Control) address is a unique number identifying a unique device or port of a device.

When you see the term “MAC address”, it means a numeric MAC address.

Numeric MAC Address

A MAC address contains six sections. Each section contains 8 bits expressed as a hexadecimal number (00–FF).

Sometimes the first three sections and last three sections are separated by one space; sometimes all sections are presented as one, without spaces; sometimes each section is separated by a colon or a space.

Examples: 010203 FDFEFF, 010203FDFEFF, 01:02:03:FD:FE:FF

MAC Address including OUI

This type of MAC address is sometimes (inaccurately) referred to simply as an OUI. In fact, the Organization Unique Identifier (OUI) comprises the first three sections of a MAC address. If DDM Inventory recognizes the numeric form of the OUI, it replaces the numbers with a short form of the organization name. This makes it easier to identify a device. If DDM Inventory uses an alphabetic short form for a device’s OUI, the device is said to have a recognized OUI. Having a recognized OUI is sometimes abbreviated to “having” an OUI.

Example: DELL 59FC91

Netmask Notation

Network masks, often referred to as netmasks, can usually be expressed in two formats in IPv4—either the familiar octet notation (also called dotted decimal notation) or CIDR notation.

Example of octet notation: 255.255.255.248

Example of CIDR notation: 29

The shorter CIDR notation is based on the binary equivalent of the octet notation, and refers to the numbers of contiguous 1s. Below are examples of netmask notation:

255.255.255.255	11111111.11111111.11111111.11111111	32 1s
255.255.255.248	11111111.11111111.11111111.11111000	29 1s
255.255.0.0	11111111.11111111.00000000.00000000	16 1s



A valid netmask contains a series of contiguous 1s. Zeros can appear after a sequence of 1s but cannot appear before or between the 1s. If zeroes appear before or between the 1s, it is not a valid netmask.

In IPv6, netmasks can only be written in CIDR notation.

Community Strings/Users

Depending on the version of SNMP supported on a device, a management system can access the SNMP MIB with a community string (SNMPv1/v2) or a user (SNMPv3).

Community strings and users are like device-based password that control access to the SNMP MIB of a device. A device controls its own community strings/users, but you must tell DDM Inventory about them.

If DDM Inventory is not given the correct community strings/users and access to devices on your network, DDM Inventory will be unable to read device MIBs. DDM Inventory will then assume that each device it cannot read has no SNMP management available.

Multiple Strings

For each device that it discovers, DDM Inventory will try all the community strings/users you have provided for that device and use the first string that receives a positive acknowledgement to read or write to the system MIB. This means that DDM Inventory may try several community strings/users before it finds one that will cause the device to respond.

The fact that DDM Inventory may try several community strings/users has implications for any devices that issue SNMP traps (also known as security traps and authentication traps).

SNMP Traps

Some devices may issue an SNMP trap when DDM Inventory attempts to explore them. Even if DDM Inventory has the correct community string/user in its list, DDM Inventory may still “trip” the trap if DDM Inventory tries multiple community strings/users before finding the right one.

For example, DDM Inventory might try two invalid community strings before reaching the valid community string. Any invalid community string will “trip” a security trap.

Once a trap has been tripped, the trap may be re-issued periodically until the trap is reset. DDM Inventory does not reset traps. Therefore, you should either disable all such traps or use only a single correct community string/user for each device that issues a trap.



If another network management system is used in the same network with DDM Inventory, this other system may generate alarms due to these traps.

Directed Community Strings

If a device is programmed with a directed community string (sometimes known as a direct access list), it will reject the attempt by DDM Inventory to SNMP QUERY it, even if DDM Inventory has been given the correct community string. With a directed community string, each device checks not only the “password,” but also to see if the DDM Inventory server is on the list of “trusted” devices.

You can allow DDM Inventory to communicate with a device with a directed community string, but you cannot do so merely by configuring DDM Inventory. You must also give the device itself an entry for a directed community string associated with the IP address of the DDM Inventory server.

Bridge Aging

To obtain the best results with DDM Inventory, turn bridge aging on. Also, set the aging interval for 2–6 hours, although some circumstances may call for an aging interval as long as 12 or even 24 hours. (Longer aging intervals are not always possible. A common maximum aging interval is 32767 seconds, or just over 9 hours.)

Bridges, routers, and switches generally have tables in which they store the addresses of devices on the network. The tables are periodically purged and relearned in order to keep the list of devices current. The aging interval defines the frequency with which tables are purged and relearned.

When there is no table entry for the address of an incoming packet, the bridge, router, or switch must learn the location of the address. To learn the location, the device sends the incoming packet to all its own ports. (This is often referred to as “flooding” or “leakage”.) When the destination device with the corresponding address responds, the bridge, router, or switch learns the location and makes an entry in the address table.

If the table is full and a new entry must be made, the “oldest” entry is usually replaced by the new entry. Device manufacturers commonly strive to include a table large enough to hold the addresses of all active sessions, but space in a table is always finite.

DDM Inventory reads the tables of bridges, routers, and switches to learn the addresses of all the connected devices. Many bridge, router, and switch vendors use a standard aging interval of 300 seconds (5 minutes), which is too short.

If the bridge aging interval is too short:

- DDM Inventory may never discover devices that are connected to the network for short periods—for example, laptops.
- DDM Inventory may take longer to determine connections between devices that it has discovered.
- Tables will be purged so frequently that flooding will occur regularly, using bandwidth unnecessarily.

If bridge aging is not turned on for a device, or if the bridge aging interval is too long:

- Tables will contain old addresses of devices that may have been removed from the network or devices that are broken. As a result, DDM Inventory will work from an outdated and possibly confused representation of what is in your network and how it is connected.

OSI Model Layers

The Open Systems Interconnection (OSI) model has seven layers. Layers 2 and 3 are the most important to DDM Inventory:

- Layer 2 is the Data Link layer, at which level MAC addresses are used. Bridges and some switches are layer 2 devices.
- Layer 3 is the Network layer, at which level IP addresses are used. Routers are layer 3 devices.

Some switches are both layer 2 and layer 3.

The seven layers are:

Layer number	Layer
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Management Workstation

Any workstation or personal computer capable of running a supported web browser. There is more detail on requirements for a management workstation in the *Installation and Initial Setup Guide*.

Virtual Machine

It is a virtualized environment that emulates in software the hardware of a real machine, including a set of emulated physical devices. Operating systems and other programs can be installed on a virtual machine and will behave as if installed on a real physical computer. The virtual machine utilizes the physical resources, such as RAM and disk space, of the physical host machine it is installed on.

Solaris Zone

It is a software partitioning technology, which provides a means of virtualizing Solaris operating system services to create isolated environments (zones) for running applications. This isolation prevents processes that are running in one zone from monitoring or affecting processes running in other zones. Different amounts of physical resources (such as RAM, CPU utilization, etc.) can be allocated to individual zones.

DDM Inventory Terms and Concepts

These terms and concepts are either unique to DDM Inventory, or have a special meaning in this context.

Object Label

For devices, the object label tells you what kind of device it is. For packages, the object label tells you how many devices are in the package.



Real Device

- device tag classifies the device
- device title identifies a specific device

Tag type	Example
Rule-specific ^a	Cisco NCD?
Model	Cisco 1601
Family	Cisco 1600
Network Function	Optivity
Operating System	Windows 95
Registered SysObjId Manufacturer	Novell Inc
Registered OUI(MAC) Manufacturer	Cisco

- a. Limited information is available, or, a managed device is not listed in the DDM Inventory Rulebase; see also the following table.

Ending	Meaning
?	less than 90% probability of identity
NCD?	DDM Inventory is relying on the MAC address. The OUI indicates that the device is probably a network connectivity device (NCD), but there is some possibility that it may be an end node.

Connector Device

- no device tag
- device title can identify a subnet or can be arbitrary

Package

- package tag shows number of devices contained by package
- package title can identify parent device (automatic package) or top object of package (multi-object package); can also be arbitrary (any package)

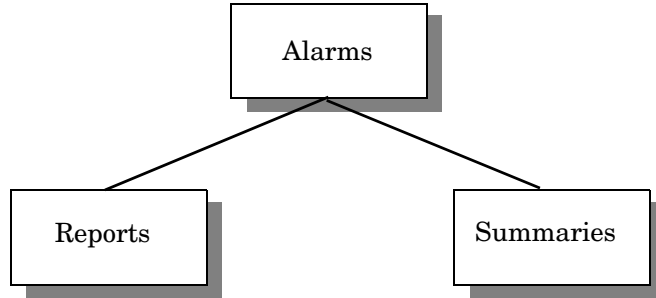
Events and Alarms

Events are caused by changes on the network, such as adding a new device, or changing a device property such as its icon or title. All of these events are reflected in the Events Browser.

There are 4 basic event types:

Event Type	What is Generated	Example
Add	Event in the Events Browser	new device added to network
Delete	Event in the Events Browser	the device is hidden or has been deactivated
Move	Event in the Events Browser	connectivity change, physically moving a device
Property Change	Event in the Events Browser	changing a device icon, priority





An event triggers an alarm. For detailed information about the specific alarms that DDM Inventory recognizes, see **Help > Classifications > Alarms**. The following diagram shows the alarm hierarchy:



Reports (for example, MTTR and MTBF) are accumulated data; they summarize data for the past 24 hours. You can change the default “time period” for these alarms in **Administration > System Configuration > MTTR and MTBF**.

Summaries (for example, Adds and Deletes) pertain to specific events. You can change the default “time period” for these alarms in **Administration > System Configuration > Adds/Deletes/Changes/Moves**.

Here is a list of the alarm indicators that are visible in the Health Panel and elsewhere in the user interface:

Alarm Type	Indicator	
n/a (not an alarm state, this indicates that the attribute is not being monitored)		blank
OK	–	dash
Info		green asterisk
Minor Alarm		gold triangle
Major Alarm		orange diamond
Critical Alarm		red square

Panel Elements

Certain elements are common to all Device Manager or Port Manager, Line Manager, or Attribute Manager panels:

- When data in a table has a gray background, the data shown is considered stale, because it was obtained before the beginning of your selected time period. (In some cases, data may be shown in parentheses rather than with a gray background.) To change the time before data is considered stale, see the section on Account Properties in the *Configuration and Customization Guide*.
- A blank space indicates that data is not available for a device or port.
- The final line on each panel is the date and time that the panel was refreshed. (This refers to rendering the panel itself, not when the data shown in the panel was last read.) This date can be useful when you print a panel. To change the format of this date, see the section on Account Properties in the *Configuration and Customization Guide*.

Banner

The banner that appears at the top of all Device Manager or Port Manager, Line Manager, or Attribute Manager panels consists of several elements.

Element	Example	Notes
Attribute Name	Total Breaks	This only appears in the Attribute Manager
Device title and IP address	website.example.com / 192.168.96.1	see Device Title on page 55 if the device title is the IP address, the IP address is shown once if there is no IP address, only the device title is shown
Manager name	Device Manager	—
System name of DDM Inventory server	ExampleCorp	see the <i>Installation and Initial Setup Guide</i>
Web browser name	Netscape Internet Explorer	—

Device Title

Each device in the DDM Inventory database can have many different pieces of identifying data. It may or may not have an IP address, a DNS name, a MAC address, an asset tag, a VM name, and so on. People have different preferences with respect to how devices are identified in the DDM Inventory user interface (UI). Some like to refer to all their network devices by their IP addresses, for example, while others prefer to use internal asset tags. You can specify which type of data is used to identify devices by customizing your device title preferences.

The device title is an identifier (name) that appears in the various windows, web pages, and reports available in the DDM Inventory UI. The device title is derived based on a priority order that you specify. You might, for example, tell DDM Inventory to use the DNS name as the device title—unless the device doesn't have a DNS name, in which case it should use the IP address instead.

Each device has only one title. The device title is an actual data field that contains text. This text is copied from the source data field that you specify.

To customize your device title preferences, go to **Administration > System Configuration > Display preferences**. Here, you can specify a prioritized list of source fields that determines the sequence in which DDM Inventory attempts to assign a title to each device. It begins with the source field at the top of your list and looks for data in that field. If it finds data, it stops there. If it does not find data, it continues to the next source field. If it does not find data in any of the fields in your list—or if your list is empty—it uses “Unknown” as the device title.

You can either use the default device title preferences list or create a customized priority list of your own. The following fields are available:

- Asset tag (scan)
- BIOS Asset tag
- Computer name (scan)
- NetBIOS name (network)
- Last name
- First name
- Device-specific title
- Domain name
- Host name
- VM name
- Operating system
- Family
- Model
- Network function
- System description
- System name
- System location
- System contact
- IPv6 address
- IPv4 address
- MAC address including OUI
- MAC address (all-numeric)
- Phone number (mobile)
- Subscriber name (mobile)



Administrator or IT Manager accounts can explicitly change the title for a particular device using the **Device Properties** button in the Device Manager. This overrides the device title preferences. Device titles are global. To determine the default title for a device, see the Diagnosis panel on the Device Manager.

4 Virtualization in DDM Inventory

Overview

Virtualization allows administrators to set up several virtual devices on one physical device. This is a popular technology, since there is often a need to consolidate physical machines as virtual machines on one physical server. This provides flexibility to relocate and add virtual machines as needed.

DDM Inventory provides you with the ability to discover, scan, and collect utilization data on virtual devices just as if they were physical devices. Some virtualization technologies offer advanced support. When this support is available, DDM Inventory can not only provide the basic discovery, inventory, and software utilization data but can also determine the parent/child relationship between the physical host device and the virtual devices hosted on the physical device. As such, you are able to see the host/virtual device information in several places in DDM Inventory including the Network Map, the Virtual Devices Window, the Device Manager, and the Virtual Device Reports.

Supported Virtualization Technologies

DDM Inventory supports two virtualization technologies that provide advanced support: VMware and Solaris zones.

VMware on the ESX server 3.0 and higher supports the Web Services interface that allows DDM Inventory to determine parent/child relationships between the physical host machine and the virtual machines hosted on the physical machine.

Solaris zones technology on Solaris 10 also provides the support that allows DDM Inventory to determine parent/child relationships between the physical host and the virtual zones hosted on the physical host.

This chapter emphasizes VMware on ESX server 3.0 and higher and Solaris zones on Solaris 10 because of the advanced support that these technologies provide. However, DDM Inventory also supports several virtual environments that provide just basic support, that is, they do not provide the information required by DDM Inventory to determine parent/child relationships. These virtual environments include VMware workstation, VMware GSX server, VMware prior to 3.0 ESX server, Virtual PC, Virtual server, IBM AIX LPAR, and HP-UX vPar. All of these environments can be detected by the scanner, and the scanner can exit if no inventory is required for these virtual environments. If inventory is required, these devices are treated like normal physical devices. As such, scanners in these virtual environments must be executed and configured to not exit in that specific virtual environment.

So while this chapter specifically cites the technologies that provide advanced support, the information that does not rely on the parent/child relationship also applies to the environments that provide only basic support.

VMware

This technology provides the ability to host several virtual machine (VM) images on one physical device. The VM image represents a virtual machine with virtual hardware and with an OS (which can be different from the host OS) and applications running on it that are distinct from the host and other VMs. In this implementation, the OS is not aware that it is running on a virtual machine.

DDM Inventory enables you to see how VMs are being used in your network environment. VMs are linked to their physical host machines, but each VM is treated as a separate device with respect to discovery, hardware scanning, and software utilization. For example, each VM appears as a distinct device on the Network Map. DDM Inventory can be configured so that an agent is deployed to the host machine and to each VM.

You can collect and view discovery, inventory, and software utilization data for individual VMs, and you can view a list of all the VMs hosted by a given server. You can configure the frequency with which DDM Inventory queries the host server to track the dynamics of the virtualization environment. You can also set up virtualization profiles that are associated with specific device groups. Refer to the “Virtualization Profiles” section in the “Configuring the Discovery Process” chapter in the *Installation and Initial Setup Guide*.

Solaris Zones

This technology provides the ability to partition a Solaris host system into zones (containers), each running a Solaris OS with applications. The host system is referred to as the global zone, and the hosted zones are referred to as local (or non-global) zones. The local zones are not aware of each other or the host system. This is a very light weight virtualization, implemented at a very high level.

DDM Inventory enables you to see how the Solaris server is being used. The Solaris zones are linked to the host server. Each zone is treated as a separate device. The host global zone physical device and the Solaris local zones appear as distinct devices on the Network Map.

As with VMware, you can collect and view discovery, inventory, and utilization data on Solaris zones and these zones are linked to the Solaris host.

Support for Virtual Devices in DDM Inventory

Discovering Virtual Devices on the Network

DDM Inventory can discover virtual devices in the same manner that it discovers physical devices on the network. It can discover virtual devices on its own by pinging and polling through a collection of IP addresses (device groups) that you provide. As it does with physical devices, after it discovers a virtual device, DDM Inventory creates a device model (a unique description of that device) and adds the created device model to the DDM Inventory database. You can prevent DDM Inventory from adding virtual devices to the database if you like. See [Virtual Devices in the DDM Inventory Database](#) on page 66.

VMware VirtualCenter

DDM Inventory supports VMware VirtualCenter 2.5 and later. VirtualCenter is virtual infrastructure management software, which provides a central point of control for the network's virtual computing resources. VirtualCenter centrally manages ESX and GSX physical servers and the Virtual Machines (VMs) that they host as a single, logical pool of resources. It manages access control globally across servers and virtual machines.

So what does this support provide in DDM Inventory? It simplifies discovery of ESX and GSX Servers and the VMs that they host. You just have to provide credentials to connect to VirtualCenter. You can then discover information about all of the host ESX and GSX Servers (and their VMs) that VirtualCenter centrally manages at one time. It allows complete discovery from a single point. Without this support, it is necessary to provide credentials for each ESX and GSX Server so that DDM Inventory can discover them and their virtual resources individually. VirtualCenter greatly optimizes discovery in a VMware virtual environment.



The administrator account can be used to access the ESX Server. If for security reasons you want to set up a specific account for this purpose, the user account used to interact with the ESX Server must have the `retrieveProperties` privilege enabled. DDM Inventory will only read properties; it will not change them.

For more information, refer to the “Privileges” appendix in the *VMware Infrastructure SDK Programming Guide* available at the VMware website.

Scanning Virtual Devices

Scanners can be distributed to individual virtual devices from the DDM Inventory server using the Agent. Agentless scanning is also possible on virtual devices just as it is on physical devices. Refer to the “Setting up Agents and Scanners” chapter in the *Installation and Initial Setup Guide*.

VMware

In the case of VMware, if you want to collect inventory data on the physical host and all the VMs hosted on the physical machine, you must execute the scanner on the hosting physical machine, as well as inside each VM hosted on the physical machine. You can scan the host physical machine and each VM for its hardware components and collect a list of the software applications installed. General host information is initially available through the remote discovery procedure using a Virtualization profile. Eventually, the scanner can be run on the host machine itself to collect more detailed information.

Scans can be scheduled to occur automatically, or they can be performed manually in the same manner as physical devices. For automatic scans, the server maintains a schedule dictating which devices should be scanned and when.

▶ Not all ESX Server machines can be scanned using the Linux Scanner. See **Help > Compatibility Matrix** for support details.

▶ When the built-in firewall on an ESX Server is enabled, access to the port that the Agent uses may be blocked. You can use the `esxcfg-firewall` command to enable access to a specific port on the ESX Server. If the Agent is configured to use port 2738, for example:

```
esxcfg-firewall -o 2738,tcp,in,discaagt
```

For more information, refer to your VMware ESX Firewall documentation.

Solaris Zones

In the case of Solaris zones, there are two possible ways to scan for data in the zones. They are the following:

- **Global zone scan:** In this scenario, the scanner on the global zone collects all hardware and software inventory for the server machine and produces a single scan file. Since the global zone has knowledge of the local zones, this global scan file can then be used to generate individual scan files for each of the local Solaris zones. See [Processing Inventory for Virtual Devices](#) on page 62. When using this method, you deploy an agent to the global Solaris zone only. This ensures that when a scanner is running in Enterprise mode that only the global zone is scanned. When running a scanner in Manual Deployment mode, you deploy the scanner to the global Solaris zone only to ensure that just the global zone is scanned. The scan file produced using this method contains hardware inventory information mainly for the global zone. The generated local zone scan files contain software inventory for the local zone and minimal hardware inventory information.

Advantage: It avoids creating additional workload on the Solaris server because the directory structure common to multiple Solaris zones is scanned only once.

Disadvantage: It contains no detailed hardware and configuration information for each local zone. If any volumes are mounted dynamically after the zone is up and running, this method may not be able to scan these additional volumes because the inventory collected in the global zone does not know about them. If each zone is a large zone in the sense that it does not inherit directories from the global zone, there is little benefit in using this mode.

- **Local zone scan:** In this scenario, an agent can be installed on each local zone. Agentless scanning is also possible. A scanner runs on each zone and produces a scan file for that local zone.

Advantage: It produces scan files with more detailed hardware and configuration information and includes software inventory even for volumes that are mounted dynamically.

Disadvantage: It creates additional workload on the Solaris server because common directories are scanned multiple times and multiple agents consume more system resources.

IBM AIX LPAR

In the case of IBM logical partitions (LPAR), an LPAR is a logical partition, which is a subset of the computer's hardware resources, virtualized as a separate computer. In effect, a physical machine can be partitioned into multiple LPARs, each housing a separate operating system. You cannot scan the physical host itself. The scanner runs inside each LPAR hosted within the physical host. If you want to collect inventory data on all of the LPARs hosted on the physical host, you must execute the scanner inside each LPAR hosted on the host machine. You can scan each LPAR for its hardware components and collect a list of the software applications installed.

Scans can be scheduled to occur automatically, or they can be performed manually in the same manner as physical devices. For automatic scans, the server maintains a schedule dictating which devices should be scanned and when.

HP-UX vPar

In the case of HP-UX virtual partitions (vPar), the physical host machine is sliced into Npars, which are hardware partitions. The vPars run inside of an Npar. Each vPar runs its own instance of HP-UX and has no knowledge of other vPars running in its hardware partition. However, the scanner is able to determine sibling relationships between vPars and parent relationships with Npars if this information is available. If you want to collect inventory data on all of the vPars hosted on an Npar, you must execute the scanner inside each vPar for that hardware partition. As with the other virtualization technologies, you can scan each vPar for its hardware components and collect a list of the software applications installed.

Scanner Options for Virtualization

In the Scanner Generator, you can indicate if you want containers included in a hardware detection scan.



Containers include Solaris Zones, HP-UX vPars, and IBM LPARs.

The **Containers** check box is an option under the **Operating System** category on the **Hardware Data** screen in the Scanner Generator GUI. Currently this option is relevant only to the Solaris operating system.



The **Containers** check box must be enabled for the global zone to recognize its parent/child relationship with the local zones that it hosts.

Also, on the **Scanner Options** screen on the **Miscellaneous** tab in the Scanner Generator GUI, you can specify that you want the scanner terminated if it is running in a virtual environment.

Scanners can detect if they are in a virtual environment. You can configure the scanner to exit if it encounters a virtual environment. By default, all virtual devices are turned on for scanning except for the non global Solaris zone. You may want to enable the termination option for this virtual environment if you plan to run the scanner in the recommended global zone mode because you do not want the scanner to run on the local zones.

Refer to the “Scanner Generator” chapter in the *Configuration and Customization Guide*.

Processing Inventory for Virtual Devices

VMware Devices

For VMware devices, individual scan files are created for the VMs hosted on the physical machine if you have configured DDM Inventory to collect this data. Inventory processing occurs in the same manner that it does for any physical machine on the network. The physical host is initially scanned remotely. It can be scanned later by running a local scanner on the physical machine to get additional information.

Solaris Zones

For Solaris zones, there is an option that allows you to specify to the XML Enricher how you want it to process the global zone scan file (See [To enable the generation of local scan files from the global scan file](#)). If this option is enabled, the XML Enricher generates individual scan files for each local zone from the global zone scan file on the Solaris server.



Enable this option if you are running the scanner in the recommended global zone scan mode. By default this option is enabled.

Disable this option if you are running the scanner in the non-recommended local zone mode. Enabling this option causes the XML Enricher to discard the local scan files produced for each local zone by the individual scanners and to replace them with scan files that are generated from the global scan file.

To enable the generation of local scan files from the global scan file

- 1 Select **Administration > System Configuration > Scan Processing**.
- 2 For the **Generate Solaris local zone inventory from the global zone** option, select **Default**.

Refer to the “Configuring the XML Enricher using the WEB UI” section in the “XML Enricher” chapter in the *Configuration and Customization Guide*.

IBM and HP-UX Partitions

For IBM AIX LPAR and HP-UX vPar, individual scan files are created for the partitions if you have configured DDM Inventory to collect this data. Inventory processing occurs in the same manner that it does for any physical machine on the network.

Also, additional scanner hardware fields are collected. As a result, several more attributes are parsed and added to the scan files on these systems providing more comprehensive information about the hardware resources used by their virtual machines.

The generic scanner hardware virtualization information (VM name, type, etc.) is stored in the same fields used for Solaris zones (hwOSContainerXXX). Information specific to LPAR and vPar are captured in a list of properties.

The properties collected for IBM AIX LPAR are the following:

- Maximum Virtual CPUs
- Unallocated Capacity
- Partition Name
- Unallocated Weight
- Online Memory
- Maximum Capacity
- Capacity Increment
- Entitled Capacity
- Variable Capacity Weight
- Active CPUs in Pool

- Active Physical CPUs in system
- Partition Group-ID
- Partition Number
- Shared Pool ID
- Node Name
- Maximum Physical CPUs in system
- Online Virtual CPUs
- Minimum Memory
- Type
- Maximum Memory
- Minimum Virtual CPUs
- Mode
- Minimum Capacity

You can find the details for these properties on the manual page for `lparstat(3)` on AIX.

The properties collected for HP-UX Npar are the following:

- CoreCellCapable
- ConnectedTo
- Cell
- UseOnNextBoot
- Cabinet
- CPU
- Memory
- NparStatus: String returned by `parstatus -C -M`

You can find the details for these properties on the manual page for `parstatus(1)` on HP-UX.

The properties collected for HP-UX vPar are the following:

- VparName
- CpuMinMax
- NPartitionID
- KernelPath
- VparStatus: String returned by `vparstatus -w -M`
- State
- CpuUnbound
- CpuBound
- KernelOpt
- BootAttributes

You can find the details for these properties on the manual page for `vparstatus(1M)` on HP-UX.

Collecting Utilization Data for Virtual Devices

In addition, you can also enable agent plug-ins to collect software utilization information on virtual devices.

For VMware, the agent must be installed (with the utilization option turned on) on the physical host machine and on each VM running on that machine.

For Solaris zones, the agent must be installed (with the utilization option turned on) on the host global zone only.

Summary of Scanning Procedures

For VMware devices:

- Install the scanner agent on each hosted VM. You can later install the agent on the physical host machine if precise inventory information is required for the host. It is also possible to perform an agentless scan on the VMs and host machine. With agentless scanning, it is not necessary to install an agent. See the “Setting Up Agents and Scanner” chapter in the *Installation and Initial Setup Guide*.
- Disable scanner termination for VMware (default setting) on the **Miscellaneous** tab of the **Scanner Options** screen in the Scanner Generator GUI so that the scanner will not terminate when it detects this environment.

For Solaris zones using global scan mode:

- Install the scanner agent on the global Solaris zone only. It is also possible to perform an agentless scan on the global zone. With agentless scanning, it is not necessary to install an agent. See the “Setting Up Agents and Scanner” chapter in the *Installation and Initial Setup Guide*.
- Enable **Containers** to be detected on the **Hardware Data** screen in the Scanner Generator so that the global zone can determine its parent/child relationship with the local zones.
- Enable scanner termination for non global zones on the **Miscellaneous** tab of the **Scanner Options** screen in the Scanner Generator GUI so that the scanner will terminate when it detects this environment.
- Enable generation of scan files for local zones from the global zone scan file in the Web UI.

For Solaris zones using local scan mode:

- Install the scanner agent on all of the local zones. It is also possible to perform an agentless scan on the local zones. With agentless scanning, it is not necessary to install an agent. See the “Setting Up Agents and Scanner” chapter in the *Installation and Initial Setup Guide*.
- Disable **Containers** to be detected on the **Hardware Data** screen in the Scanner Generator since parent/child relationship information is not needed.
- Disable scanner termination for non global zones on the **Miscellaneous** tab of the **Scanner Options** screen in the Scanner Generator GUI so that the scanner will not terminate when it detects this environment.
- Disable generation of scan files for local zones from the global zone scan file in the Web UI so that the local scan files produced by the scans on local zones are not discarded by the XML Enricher.

For Microsoft Virtual PC/Virtual Server:

- Install the agent on the host machine and on each hosted virtual device. It is also possible to perform an agentless scan on the virtual devices and host machine. With agentless scanning, it is not necessary to install an agent. See the “Setting Up Agents and Scanner” chapter in the *Installation and Initial Setup Guide*.
- Disable scanner termination for Virtual PC (default setting) on the **Miscellaneous** tab of the **Scanner Options** screen in the Scanner Generator GUI so that the scanner will not terminate when it detects this environment.

For Windows Terminal Services:

- Install the agent on the host machine only. It is also possible to perform an agentless scan on the host machine. With agentless scanning, it is not necessary to install an agent. See the “Setting Up Agents and Scanner” chapter in the *Installation and Initial Setup Guide*.
- Disable scanner termination for Terminal Services (default setting) on the **Miscellaneous** tab of the **Scanner Options** screen in the Scanner Generator GUI so that the scanner will not terminate when it detects this environment. If you use Manual Deployment mode to run scanners (for example, via login scripts), you must enable scanner termination for Terminal Services in order to prevent the scans from occurring in multiple terminal sessions.

Virtual Devices in the DDM Inventory Database

You can specify whether you want DDM Inventory to add virtual-only devices, such as Solaris zones and VMware VMs, to the database.

To prevent virtual devices from being added to the database

- 1 Select **Server > Administration > System Configuration > Input Filters**.
- 2 Under **Devices not to add to the database**, select **Custom** and then check the **Virtual-only devices** option.

DDM Inventory treats a virtual device (zone or VM) that has an SNMP agent installed as a managed device. It treats a virtual device with a manual scanner installed, but no SNMP agent, as a scanned device.

Virtual Devices on the Network Map

By default, individual virtual devices such as Solaris zones or VMware virtual machines are displayed on the Network Map. You can choose not to have these devices appear on the Network Map if you prefer.

To prevent virtual devices from appearing on the Network Map

- 1 Select **Server > Administration > System Configuration > Network Devices**.
- 2 For the **Show virtual devices on the Network Map** option, select **No**.

Regardless of this setting, you can use the Virtual Devices Window to display a list of the virtual devices associated with a particular host. See the next section.

Virtual Devices Window

Once you locate a network device through the **Find** command, the Network Map, or any other method, there is an option on the **Device** menu in the various applet windows that allows you to see if the device is a virtual device, a machine hosting virtual devices, or a VMware VirtualCenter device that manages VMware hosts. If the device falls into one of these categories, the **Show Virtual Devices** option will be enabled on the **Device** pull-down menu. Selecting this option opens the **Virtual Devices** Window.

For a virtual host or virtual device, this window displays the server type, its version, platform, and model, with the number of virtual devices that it hosts. Below this, the following information is displayed for each virtual device associated with the host device:

- **VM:** Logical machine name of the virtual device
- **IP Address:** IP address of the virtual device
- **VM Name:** Name assigned to the virtual device
- **VM OS:** Operating system running on the virtual device
- **VM Status:** Current status of the virtual device
- **Update Time:** Time that DDM Inventory last collected information about this device

If the device is a VirtualCenter server, this window displays server type and platform, with the number of virtual hosts that it manages. Below this, the following information is displayed for each VMware host managed by VirtualCenter:

- **VM Host:** Logical machine name of the VMware host
- **IP Address:** IP address of the VMware host
- **Server:** Type of the virtualization software running on the VMware host
- **Version:** Version of the virtualization software running on the VMware host
- **Platform:** Operating system running on the VMware host
- **Model:** Model of the VMware host machine

If you double-click on any of the devices in **Virtual Devices** window, the device opens in the Device Manager. You can also access the Device Manager and the Network Map by the icons provided in the upper right-hand corner of the window.

Virtual Devices and the Device Manager

The Device Manager displays specific items when it knows that a device is a virtual device, a machine that hosts virtual devices, or a VirtualCenter server that manages VMware host machines.

For VMware hosts, the Configuration panel includes the VMware Credentials table. This table displays the preferred VMware credentials, including the user name and a password hint, for this VMware host. Preferred credentials are the credentials that were used most recently to successfully retrieve information from the VMware host machine.

The VMware Credentials table is present only for VMware host devices. The information in the table is populated after the VMware discovery process is completed for this host.

The Diagnosis panel contains three items that pertain to virtualization.

- The first item is the Configuration Profiles table, which shows the name of the virtualization profile that applies to this device.

- The second item is the Discovery Configuration table, which lists three virtualization parameters for this device: VMware discovery credentials, VMware discovery schedule, and VMware discovery interval. These parameters are specified in the virtualization profile that is associated with this device.
- The third item is the Virtualization Log button on the Diagnosis panel toolbar that displays information logged during the VMware discovery process running on the host.

The Update Model panel contains the **Run VMware Discovery** option in its pull-down menu that allows you to force the discovery of a device that is a VMware host (ESX server 3.0 or later) so that its network model and the model of its hosted VMs are updated immediately in the DDM Inventory database. The **Run VMware Discovery** option is also available from any applet window from the right-click menu associated with a device.

For additional information, refer to the “Using the Device Manager” chapter in the *Network Data Analysis Guide*.

Virtualization Configuration Profiles

A virtualization profile enables you to specify two things: (1) VMware credentials, and (2) how often and when the discovery process for virtual devices is initiated. A virtualization profile can be associated with one or more device groups. A device group can be organized either by IP range or device type.

For additional information about discovery configuration profiles, refer to “Configuring the Discovery Process” in the *Installation and Initial Setup Guide*.

Reports on Virtual Devices

You can see a summary and detailed reports of all hardware inventory data collected for Solaris zones and VMware.

To view the virtual devices summary report

- 1 Select **Server > Reports**.
- 2 Under **Virtualization Reports**, click **Virtual Devices**. You can click the link of the virtualization type report you want to see.

Refer to the “Using the Reports” chapter in the *Network Data Analysis Guide*.

Deactivation and Purging of Virtual Devices

You can indicate when you no longer want virtual-only devices included in your reports or in the DDM Inventory database.

You can specify a deactivation interval for a virtual-only device. At the end of this deactivation interval, the virtual device is removed from the Reports, but its data is kept in the DDM Inventory database. If DDM Inventory sees that device again, or if you manually “reactivate” the device, it will reappear on the Reports.

You can also specify a purge interval for a virtual-only device. At the end of this purge interval, the virtual device is removed from the DDM Inventory database. The purge interval starts at the end of the deactivation interval. The virtual device’s deactivation interval is dependent on the virtualization discovery interval for the device. The time it takes to deactivate a virtual device is either the virtual devices deactivation interval, or three times

the virtualization discovery interval, whichever is longer. For example, if you change the virtual devices deactivation interval to one day, but do not change the virtualization discovery interval of 2 days, DDM Inventory will take 6 days (3 x 2 days) to deactivate an unseen virtual device.

At the end of the virtual devices purge interval, the “deactivated” virtual device and all its data are deleted from the DDM Inventory database. A purged device may be rediscovered if it is still connected, but it will be considered a new device. A virtual device may be purged before the end of the specified virtual devices purge interval. The storage area for deactivated devices has limited capacity (10% of the device license). Once the number of deactivated devices reaches capacity, some devices will be automatically purged.

You can specify deactivation and purge intervals for a device on the **Server > Admin > System Configuration > Expiry** page.

5 Mobile Devices in DDM Inventory

This chapter describes how DDM Inventory collects, stores, and displays information about mobile devices. The following topics are covered:

- [Integration with Other HP Software Products](#) on page 72
- [Overview](#) on page 72
- [Mobile Configuration Profiles](#) on page 74
- [Enabling and Disabling Mobile Discovery](#) on page 74
- [Using a Secure Connection for Mobile Discovery and Inventory](#) on page 75
- [Mobile Devices in the DDM Inventory Database](#) on page 75
- [Protecting Private Information for Mobile Devices](#) on page 75
- [Finding Mobile Devices](#) on page 76
- [Mobile Devices and the Device Manager](#) on page 77
- [Reports](#) on page 79
- [Deactivating and Purging Mobile Devices](#) on page 79
- [Viewing Your Current Mobile Profile Settings](#) on page 80



In this chapter, the term “mobile device server” refers to the HP Enterprise Mobility Suite (EMS) server. Other types of mobile device servers are not currently supported.

Integration with Other HP Software Products

DDM Inventory integrates seamlessly with the HP Enterprise Mobility Suite, server and client software that provides comprehensive mobile device management capability. The Enterprise Mobility Suite enables enterprise IT to simplify wireless device deployments and management and to launch advanced mobile services that are secure and reliable. It conforms to standards developed and endorsed by the top handset manufacturers worldwide and features over-the-air (OTA) mobile phone initial setup, configuration, diagnostics, and security.

For more information, refer to the HP Enterprise Mobility Suite information available at www.hp.com.

Overview

DDM Inventory can discover and collect detailed inventory information about mobile devices that are managed by one or more mobile device servers in your network. Mobile devices include mobile phones, personal digital assistants (PDAs), smart phones, and other types of wireless hand-held mobile devices.

DDM Inventory discovers mobile devices differently than it does other devices in your network. To discover mobile devices, it simply queries the mobile device server that manages these devices. After it discovers a mobile device, DDM Inventory performs an inventory operation to collect detailed information about that device from the mobile device server. Additional information is then derived using data from the Rulebase. At this point, all or a subset of the following information can be displayed in the Device Manager for this mobile device:

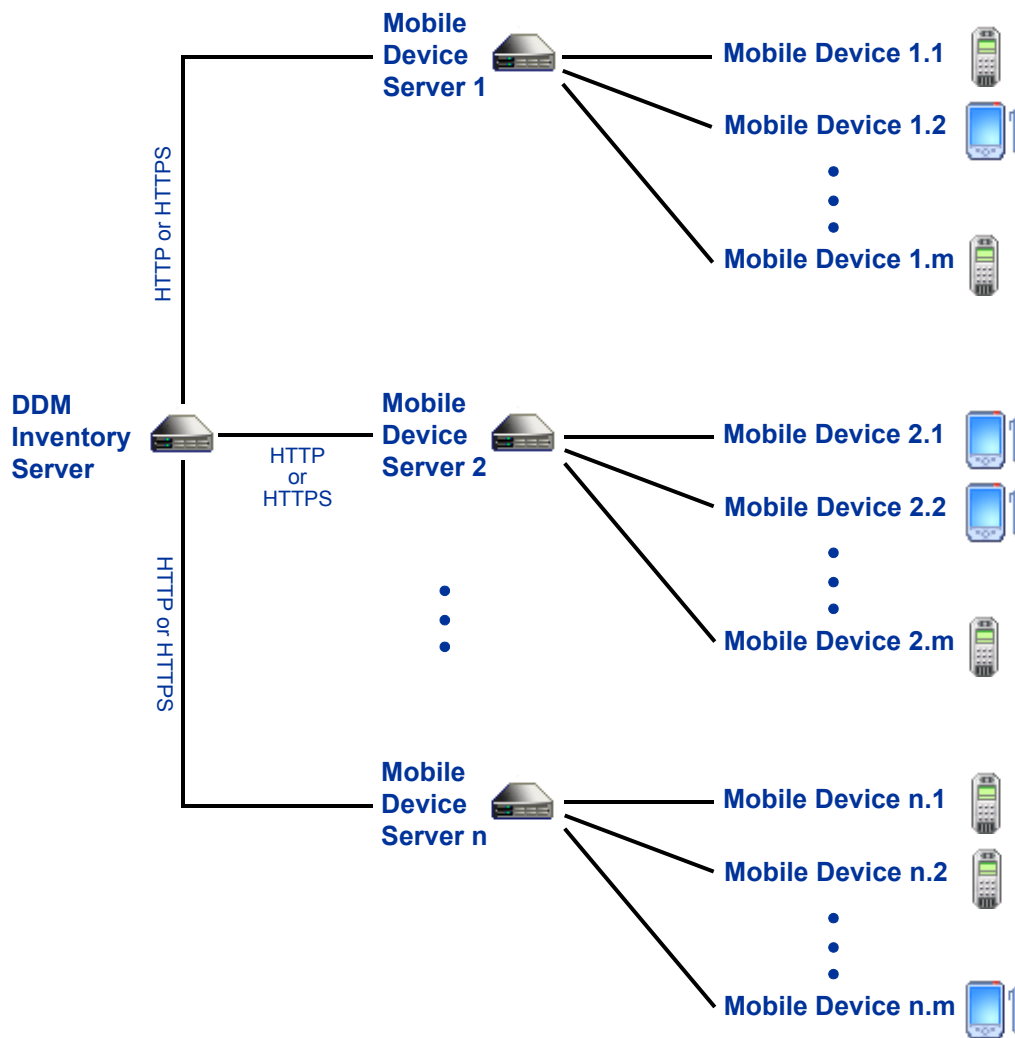
- UNSPSC description
- Model
- Model current manufacturer
- Model historical manufacturer
- UNSPSC model
- Mobile phone number
- Mobile device identifier (IMEI code, for example)
- Mobile carrier company
- Mobile carrier network
- Mobile status
- Mobile user name
- Mobile user e-mail address
- Provision date
- Type of device (mobile phone or PDA)
- Vendor (manufacturer)
- Firmware version
- Software

- CPU
- ROM
- Internal RAM capacity
- External RAM capacity
- Virtual memory



The mobile user name and e-mail address are only collected and stored if you have explicitly granted permission to DDM Inventory to collect this information. See [Protecting Private Information for Mobile Devices](#) on page 75.

An DDM Inventory server can work with many mobile device servers. The number of mobile devices that a single DDM Inventory server can discover depends on the scope of your Automated Inventory license. The maximum is 60,000 mobile devices.



Before it can discover individual mobile devices, DDM Inventory must first discover the mobile device server.



In this chapter, the term “mobile device server” refers to the HP Enterprise Mobility Suite (EMS) server. Other types of mobile device servers are not currently supported.

After DDM Inventory discovers a mobile device, it creates a device model (a unique description of that device) and adds that device model to the Discovery database. You can prevent DDM Inventory from adding mobile devices to this database if you like. Refer to [Mobile Devices in the DDM Inventory Database](#) on page 75 for more information.

The more detailed inventory information about each mobile device is stored in the Reports database. This information is used to produce mobile device summary reports. Refer to [Reports](#) on page 79 for more information.

Mobile Configuration Profiles

You can create a Mobile configuration profile that specifies when and how often DDM Inventory discovers and collects detailed inventory information about mobile devices. Mobile profiles also specify the port and the type of connection, HTTP or HTTPS, that is used for communication between the DDM Inventory server and your mobile device servers. Mobile profiles also include logon credentials for mobile device servers.

You must then associate that profile with a device group that contains your mobile device server. This can be either an IP-only or a device-type device group. To avoid unnecessary network queries, create a device group that contains *only* your mobile device server.

You can also issue a command from the Device Manager to immediately start the mobile discovery process for a particular mobile device server or collect detailed inventory information about a particular mobile device. For more information about these commands, refer to the “Update Model” section in the “Using the Device Manager” chapter of the *Network Data Analysis Guide*.

For more information about Mobile profiles and device groups, refer to “Configuring the Discovery Process” in the *Installation & Initial Setup Guide*.

Enabling and Disabling Mobile Discovery

You can turn mobile discovery on and off by setting an option in the DDM Inventory web UI. This is useful, for example, if you want to stop the mobile discovery process temporarily but you do not want to modify your configuration profiles or device group associations. By default, mobile discovery is enabled.

If the **Mobile discovery active** option is set to **No**, DDM Inventory will not discover mobile devices, regardless of your Mobile configuration profile settings. Use this override only for advanced diagnostic purposes, however. The recommended method for configuring the discovery of mobile devices is through the Mobile configuration profile settings.

To disable mobile discovery:

- 1 Select **Server > Administration > System Configuration > Discovery services**.
- 2 For the **Mobile discovery active** option, select **Custom**, and then select **No**.



It is recommended that you change this setting only when advised to do so by your HP Software Support representative.

Using a Secure Connection for Mobile Discovery and Inventory

You can use either HTTP or HTTPS protocol for communication between the DDM Inventory server and your mobile device servers. If this communication will happen over a non-secure network, be sure to select the **Use HTTPS to connect to mobile server** option in your Mobile configuration profile (or profiles). If this option is not selected, HTTP protocol is used.

For more information, refer to the “Mobile Profiles” section in the “Configuring the Discovery Process” chapter of the *Installation & Initial Setup Guide*.

Mobile Devices in the DDM Inventory Database

Provided that mobile discovery is enabled (see [Enabling and Disabling Mobile Discovery](#) on page 74), DDM Inventory adds mobile devices to its discovery database by default. You can, however, instruct DDM Inventory not to add these devices to the database by using an input filter.

To prevent mobile devices from being added to the database:

- 1 Select **Server > Administration > System Configuration > Input Filters**.
- 2 For the **Devices not to add to the database** option, select **Custom**, and then select **Mobile devices**.

➤ If you tell DDM Inventory not to add mobile devices to the database, mobile discovery is automatically disabled regardless of the value of **Mobile discovery active**.

Protecting Private Information for Mobile Devices

If you choose to add mobile devices to the DDM Inventory database, you can prevent certain information about each mobile device from being collected. This may be important to protect the privacy of mobile users. By default, the names and e-mail addresses of mobile users are not collected.

To specify mobile device information that you do not want to collect:

- 1 Select **Server > Administration > System Configuration > Network devices**.
- 2 Under **Do not collect the following mobile device information**, select **Custom**.
- 3 Select the items that you do not want to collect from the mobile device server.
- 4 Click **Change**.

➤ These settings are only meaningful if mobile discovery is activated (refer to [Enabling and Disabling Mobile Discovery](#) on page 74) and mobile devices are currently being added to the database (refer to [Mobile Devices in the DDM Inventory Database](#) on page 75).

If you change this setting, the DDM Inventory database will be updated to reflect your new setting the next time that mobile device models are updated. Refer to the description of “Mobile Profiles” in the *Installation & Initial Setup Guide* for more information about

specifying the mobile device inventory interval. Also refer to “Updating Device Models for Multiple Devices” in the *Network Data Analysis Guide* for information about manually scheduling device model updates for mobile devices.

Mobile Devices on the Network Map

Individual mobile devices are not visible on the Network Map. Mobile device servers, however, are visible.

Scanning Mobile Devices

Because mobile devices are managed by a mobile device server, you cannot scan them as you would scan other devices in your network. You can, however, collect detailed inventory information about each mobile device that has been discovered, and you can create reports that summarize this information for groups of devices. DDM Inventory obtains detailed information about a particular mobile device by querying the mobile device server that manages that device. Refer to “Mobile Profiles” in the *Installation & Initial Setup Guide* and [Reports](#) on page 79 of this document for more information.

Finding Mobile Devices

There are two ways to find individual mobile devices and get more information about them. If a device is listed on a mobile device report, you can click the device title in the report and open the Device Manager for that device. You can also find a specific mobile device by using the **Find** tool. The Basic Match feature enables you to find a mobile device by either its phone number or user name. Mobile devices are represented by the following icons:



Mobile phone



Personal digital assistant (PDA)

If the mobile device type cannot be determined, the question mark (?) icon is used to represent the device.

You can also use the Easy Find feature to find a mobile device if the phone number or the user name are part of the Device Title Preference list under **Administration > System Configuration > Display Preferences**. By default, the phone number is part of this list, and the user name can be added to the list.

For detailed information, refer to “Finding Your Network Devices” in the *Network Data Analysis Guide*.

Mobile Devices and the Device Manager

The Device Manager displays specific items when it determines that a device is either an individual mobile device or a mobile device server.

For additional information about the Device Manager, refer to “Using the Device Manager” in the *Network Data Analysis Guide*.


Mobile Device Servers

Mobile device servers look similar to other servers in the Device Manager. In addition, the Configuration panel includes the Mobile Credentials table. This table displays the preferred logon credentials, including the user name and a password hint, for this server. Preferred credentials are the credentials that were used most recently to successfully retrieve information from this mobile device server.



The Mobile Credentials table is present only for mobile device servers. The information in the table is populated only after the mobile discovery process is completed for a particular server.

The Diagnosis panel contains the following special items for mobile device servers:

- The Mobile Log () button is visible on the toolbar. This button displays information logged during the mobile discovery process.
- The Configuration Profiles table shows the name of the Mobile profile that applies to this mobile device server.
- The Discovery Configuration table shows the values of the six mobile configuration parameters: mobile discovery interval, mobile inventory interval, schedule for mobile discovery, mobile port number, mobile connection type, and mobile credentials. These parameters are specified in the Mobile profile that is associated with this device.

The drop-down menu on the Update Model panel includes commands that pertain to mobile discovery:

- The **Run Mobile Discovery** command applies only to mobile device servers.
- The **Query Device** command applies to both mobile device servers and individual mobile devices but has a different meaning in each case.

Both commands are described in detail in the following table:

Command	Description
Run Mobile Discovery	<p>Regardless of the schedule you have defined in the Mobile configuration profile, DDM Inventory immediately tries to connect to this mobile device server. If it successfully connects, it retrieves a list of the mobile devices managed by this server.</p> <ul style="list-style-type: none"> • If any new mobile devices are found, they are immediately scheduled for a model update—provided that the “Mobile inventory interval” is not set to zero in the Mobile configuration profile associated with the device group to which this mobile device server belongs. • If some existing mobile devices are found to have been deleted on the mobile device server, they are automatically deactivated in the DDM Inventory database. <p>This command is not available if the “Mobile discovery interval” is set to zero in the Mobile configuration profile for the device group to which this mobile device server belongs. By default, all device groups are assigned the <default> Mobile configuration profile. This effectively disables mobile discovery. If you want to run mobile discovery, you must create a new Mobile profile with a non-zero “Mobile discovery interval.”</p> <p>If the “Mobile discovery active” option on the Administration > System Configuration > Discovery services page is disabled, you will not be able to schedule mobile discovery. In this case, the Run Mobile Discovery command is available but has no effect.</p> <p>In addition to the drop-down list in the Update Model panel, this command is available from any applet window in the right-click menu associated with any mobile device server.</p> <p>If you select this command and do not see the results you expect, be sure to click the Mobile Discovery Log button on the Diagnosis panel toolbar.</p>
Query Device	<p>For a mobile device server, this command starts a regular network device modeling procedure. This results in a series of SNMP queries.</p> <p>For an individual mobile device, this command initiates a mobile inventory operation. When you select it, DDM Inventory determines which mobile device server manages this mobile device and issues an inventory command to that server. The mobile device server returns detailed information about this particular mobile device, such as the telephone number, manufacturer, model, operating system, user name, and so on.</p>



In addition to the drop-down list in the Update Model panel, the **Run Mobile Discovery** command also appears in the right-click shortcut menu for a mobile device server. Similarly, the **Query Device** command appears in the shortcut menu for both mobile device servers and individual mobile devices.

For additional information, refer to “Using the Device Manager” in the *Network Data Analysis Guide*.

Individual Mobile Devices

For individual mobile devices, the Configuration panel displays detailed information about the device. See the [Overview](#) on page 72 for a comprehensive list of the items that can be displayed. DDM Inventory retrieves as many items as it can from the mobile device server for each mobile device. Not all items are available for every mobile device, however.

The Diagnosis panel contains the following special items for mobile devices:

- Time this mobile device was first discovered by DDM Inventory.
- Time this mobile device was last modeled as a mobile device.
- Time this mobile device was created on the mobile device server.
- Time this mobile device was last profiled on the mobile device server.

In the Update Model panel, the Query Device command runs a mobile inventory for this mobile device.

Reports

You can create the following reports about mobile devices in your network:

Report Name	Description
Summary By Vendor, Model, Firmware Version	Device counts by vendor, model, and firmware version with drill down to device details
Summary By Mobile Carrier	Device counts by mobile carrier with drill down to device details
Locked Mobile Devices	Locked and/or wiped mobile devices

These reports are available on the **Reports > Mobile Device Reports** page. For additional information, refer to the “Using the Reports” chapter of the *Network Data Analysis Guide*.

Deactivating and Purging Mobile Devices

You can indicate when you no longer want mobile devices to be included either in your reports (deactivate) or in the DDM Inventory database (purge).

You can specify a deactivation interval for mobile devices. At the end of this deactivation interval, any mobile device that has not been seen for the duration of the interval is removed from the Reports, but its data is kept in the DDM Inventory database. If DDM Inventory sees that device again when it next queries the mobile device server that manages it—or if you manually “reactivate” the device—it will once again appear on the Reports.

You can also specify a purge interval for mobile devices. The purge interval starts at the end of the deactivation interval. At the end of the purge interval, any mobile device that still has not been seen is removed from the DDM Inventory database.

The mobile device deactivation interval is dependent on the mobile inventory interval for the device. The time it takes to deactivate a mobile device is the longer of the following:

- The mobile devices deactivation interval
- Three times the mobile inventory interval

For example, if you change the mobile devices deactivation interval to one week but do not change the mobile inventory interval of 2 weeks, DDM Inventory will take 6 weeks (3 x 2 weeks) to deactivate an unseen mobile device.

At the end of the mobile devices purge interval, the “deactivated” mobile device and all its data are deleted from the DDM Inventory database. A purged device may be rediscovered if it is still active in the network, but it will be considered a new device.



A mobile device may be purged before the end of the specified mobile devices purge interval. The storage area for deactivated devices has limited capacity (10% of the device license). Once the number of deactivated devices reaches capacity, some devices will be automatically purged.

You can specify deactivation and purge intervals for mobile devices on the **Administration > System Configuration > Expiry** page.

Viewing Your Current Mobile Profile Settings

You can view all discovery configuration settings that have been activated in table format by selecting the following item in the left navigation tree:

Status > Current Settings > Discovery configuration

This page contains a series of tables that reflect the settings that were most recently configured on the **Administration > Discovery Configuration** pages. Only changes that have been activated are reflected in these tables.

The Mobile Profiles table shows you the following settings for each activated Mobile configuration profile:

- Mobile discovery interval
- Mobile inventory interval
- Discover mobile devices using this schedule
- Mobile port number
- Mobile connection type
- Mobile credentials

6 Recorded Events

The Health Panel summarizes changes to the network. Each device should contribute only a single alarm to the Health Panel. If there is more than one alarm per device or per port, these additional alarms are displayed in the Device Manager or Port Manager.

To see a comprehensive list of alarms that are raised by report data, see **Help > Classifications > Alarms**.

Port Adds and Deletes

Identifies ports recently added to or deleted from a device. (An added port may or may not be recently discovered.)



Does not include ports on connector devices.

Port Changes

Changing interface rate, interface type, duplex, or line alarm type.

Device Adds and Deletes

Identifies devices recently added to or deleted from the database. (An added device may or may not be recently discovered.)

Device Changes

Changing icon, priority, title, or tag of a device.

Exceptions

Devices with exceptions. See **Help > Classifications > Exceptions**.

Not Recently Seen

There are two types of “not recently seen” events:

- Network Not Recently Seen
- Scan Not Recently Seen

“Network Not Recently Seen” devices are those with which DDM Inventory has lost contact and which may soon disappear from the database.

➤ Once DDM Inventory has not had contact with a device for a period greater than the threshold (by default, 6 hours), it will appear as “Not Recently Seen.” it will be displayed with a green ring. Once the “not seen” period has exceeded 24 hours, the device will also be appear faded.

“Scan Not Recently Seen” devices are devices for which DDM Inventory has not received an updated scan file (by default, 4 weeks and 2 days).

➤ You can change these defaults at **Administration > System Configuration > Report time periods.**

➤ This does not apply to connector devices

7 Scanners

The scanner used to scan each computer can capture any or all of the following types of information, depending on the options selected when the scanner was configured:

- Information about the hardware configuration.
- Information about the system configuration.
- Information about the software on the drives scanned.
- Information about the physical assets and user details that are recorded using the asset questionnaire.

The information collected by a scanner is stored in a Compressed XML File (XSF) file. This information can be viewed immediately with the Analysis Workbench or Viewer, but it can also be enriched using the XML Enricher, and have its data sent to the DDM Inventory server database. From there, the data can be viewed through the Scan Data Viewer, Reports, and so on.

Scanner Types

Scanners can be generated for the several operating systems. See **Help > Compatibility Matrix** in the DDM Inventory GUI for a complete list.

The procedure for starting a scanner depends on the native operating system environment for the computer being scanned.

Viewing the Results of the Scan

HP DDM Inventory comes with the Viewer program, which allows you to look at the results of your scans. Refer to the “Viewer” Chapter in the *Scan Data Analysis Guide* for more information about how to use this application.

For XSF scan files, a tool such as gzip or Winzip can be used to extract the XML data contained in them. The XML file contained inside the XSF file can be viewed with any text editor or XML viewer such as Internet Explorer.

Command Line Parameters and Switches

Although the options for the scanner are normally set using the Scanner Generator, it may be necessary to change some settings to allow better operation on some machines. The operation of a scanner can be modified with the use of the various command line parameters.

Reasons for Overriding the Options in a Configured Scanner

- The scanner may encounter a problem with a particular piece of hardware. Using command line options, the problem hardware can be circumvented.
- Command line parameters can change the configured options such as save path. This allows the scan results to be saved to a local machine without a full network path having to be defined.

How to Use a Command Line Parameter

You can specify command line parameters and switches by:

- Typing the command from a command line (for example, the Windows command prompt, or the UNIX/Mac OS X shell). In UNIX/Mac OS X make sure you specify the path to the scanner.

For example:

```
/tmp/scanlinux-x86 -?
```

launches the Linux scanner from the /tmp directory and displays a list of valid command line options.

- Creating a Windows shortcut. Type the command line options (if any) after the quotation marks.

For example:

```
"C:\Program Files\Hewlett-Packard\DDMI\7.60\Scanner  
Generator\Scanwin32-x86.exe" -?
```

launches the Win32 scanner and displays a list of valid command line options.

- Typing the command in the Windows Run command in the Start menu. Type in or navigate to the location where the scanner executable is located. Type the command line parameter or switch after the quotation marks.

For example:

```
"C:\Program Files\Hewlett-Packard\DDMI\7.60\Scanner  
Generator\Scanwin32-x86.exe" -?
```

Command Line Parameters for Scanners

Valid command line parameters for the scanners are shown in the following table:

Table 5 Command line parameters for scanners

Command Line Parameter	Function
-force	<p>Do not check disk space when saving offsite Scan File.</p> <p>This may be useful in situations where the operating system reports insufficient space, but this is actually due to access rights.</p>
-p:<path>	<p>Override default offsite save path.</p> <p>A UNC path can also be entered as the argument to this option. The format for a UNC path is:</p> <pre>\\servername\sharename\path\</pre> <p>For example:</p> <pre>Scanwin32-x86 -p:\\Hewlett-Packard\ED\scanfiles\</pre> <p>The user running the scanner must have Write permissions to the specified UNC path.</p>
-r:<path>	<p>Override the default path to the original scan files.</p> <p>A UNC path can also be entered as the argument to this option. The format for a UNC path is:</p> <pre>\\servername\sharename\path\</pre> <p>For example:</p> <pre>Scanwin32-x86 -r:\\Hewlett-Packard\ED\scanfiles\</pre> <p>The user running the scanner must have read permissions to the specified UNC path.</p>

Table 5 Command line parameters for scanners

Command Line Parameter	Function
<code>-scandays:<Count></code>	<p>Scan only if previous scan was more than Count days ago. Forces the scanner to perform the scan only if the previous scan was <Count> or more days ago. For example:</p> <pre data-bbox="597 457 764 485">-scandays:7</pre> <p>For example, if the scanner is launched from a login script every day, it will only perform the scan every week.</p> <p>When the <code>scandays:<Count></code> parameter is specified, the scanner attempts to check when the last scan was run. If no previous scan file is found, no messages are displayed and the scan runs.</p> <p>If a scan file is found, the following message is added to the log file:</p> <pre data-bbox="597 793 1117 821">"Checking the age of Scan File "%s"</pre> <p>Where <code>%s</code> is the full name of the scan file it checks.</p> <p>If there is a problem determining the age of the scan file (for example, if it is a newer version or it is corrupt), it then outputs:</p> <pre data-bbox="597 1024 1279 1052">The age of the Scan File cannot be determined.</pre> <p>If it does manage to obtain the date, it outputs:</p> <pre data-bbox="597 1144 971 1171">Last scan was %d days ago</pre> <p>Where <code>%d</code> is an integer number.</p>

Table 5 Command line parameters for scanners

Command Line Parameter	Function
<code>-scandayofweek:<Number></code>	<p>Scan only on specified day of week(0-Sun,1-Mon, etc.). <Number> can be one of the following:</p> <ul style="list-style-type: none">0-Sunday1-Monday2-Tuesday3-Wednesday4-Thursday5-Friday6-Saturday <p>For example:</p> <pre>-scandayofweek:5</pre> <p>This will cause the scan to be performed on Fridays only. The <code>scandays:</code> and <code>scandayofweek:</code> options can be combined. For example:</p> <pre>Scanwin32-x86 -scandays:14 -scandayofweek:3</pre> <p>This causes the scan to be performed every other Wednesday.</p>
<code>-incl:<switch></code>	<p>Switches for re-enabling individual hardware tests that were disabled in the Scanner Generator. See the Table on page 92. To include tests 10, 20 and 50, you would run:</p> <pre>-incl:10 -incl:20 -incl:50</pre>
<code>-excl:< switch ></code>	<p>Switches for disabling individual hardware tests. See the Table on page 92. To exclude tests 10, 20 and 50, you would run:</p> <pre>-excl:10 -excl:20 -excl:50</pre>

Table 5 Command line parameters for scanners

Command Line Parameter	Function
-paths	<p>Using this switch, it is possible to define exactly which directories to scan; the parameter can be repeated as many times as necessary. For example:</p> <pre data-bbox="589 451 1201 483">scan -paths:/etc -paths:/var -paths:/bin</pre> <p>will scan just /etc, /var and /bin and their subdirectories.</p> <p>Note: You must ensure that the Allow Command Line Override option is checked in the Scanner Generator Software Data tab for this to work.</p>

Table 5 Command line parameters for scanners

Command Line Parameter	Function
-o:<filename>	<p>Takes the offsite scan file name from the command line. For example (non UNIX):</p> <pre>Scanwin32-x86 -o:r:\results\SC002154</pre> <p>Where <code>r:\results\SC002154</code> is the path to the file <code>SC002154</code>. If a file name is not entered, the file is named <code>Default.xsf</code>. If the path is not specified, the file is placed in the directory configured for offsite scan files in the Scanner Generator (see the <i>Configuration and Customization Guide</i>).</p> <p>If the path is specified on the command line (even if it is relative), it replaces the path configured in the Scanner Generator. Here are some examples.</p> <pre>scanlinux-x86 -o:newname</pre> <p>Saves the offsite scan file, <code>newname.xsf</code>, to the location configured in the Scanner Generator.</p> <pre>scanlinux-x86 -o:/tmp/newname</pre> <p>Saves the offsite scan file to <code>/tmp/newname.xsf</code>.</p> <pre>scanlinux-x86 -o:subdir/newname</pre> <p>Saves the offsite scan file, <code>newname.xsf</code>, to the <code>subdir</code> subdirectory of the current directory.</p>
-log:<level>	<p>Specifies the level of debugging information that will be written to the scanner log when the scanner is running. The log is saved in the scan file and also as a separate file:</p> <ul style="list-style-type: none"> • In most cases, you can view the scanner log by using the Viewer. • If a problem has occurred that prevents the scanner from saving the scan file, you can view the scanner log file from the Diagnostic panel in the Device Manager. <p><level> can be one of the following:</p> <p><code>off</code>: Detailed logging is turned off. This is the default.</p> <p><code>debug</code>: Debug messages are logged in addition to the regular scanner messages. They are more-detailed providing additional information.</p> <p><code>trace</code>: All regular, debug, and detailed messages are logged. The detailed messages provide tracing details as to scan runs, returned error codes, and software scanning. This option automatically enables the generation of the error log file.</p>

Table 5 Command line parameters for scanners

Command Line Parameter	Function
-?	The full list of command line options can be obtained by running the scanners with the -? or /? command line option.

Viewing Command Line Options in Viewer or Analysis Workbench

If a command line option or switch has been used, it can be viewed in Analysis Workbench or Viewer.

This can be very useful when you want to check if the scan results were obtained from a scanner that had been run with any special command line options.

For example, if the scanner had been run with the -paths command:

```
scan -paths:/etc -paths:/var -paths:/bin
```

The -paths command line option will be displayed in Viewer (System Data folder in the Hardware and Configuration tab page).

Using Command Line Switches to Enable and Disable Specific Hardware Tests

Hardware test numbers that can be used for enabling/disabling hardware tests in the scanners as part of the -excl and -incl command line switches are shown in the following table:

Table 6 Hardware Tests to be used with -excl and -incl switches to Enable and Disable specific hardware tests

Hardware Test	Hardware Test
10 : BIOS Data	11 : BIOS Extension
12 : SMBIOS Information	13 : Compaq Asset Tag
14 : Plug and Play Version	30 : Video data
31 : Monitors	40 : Port data
50 : Keyboard and Mouse data	60 : Disk data
62 : Local USB hard drives	70 : Memory Data
72 : Swap Files	80 : CPU Data
90 : Operating System Data	91 : Device driver files
92 : Cluster Data	93 : Services
94 : Virtual Machine Data	95 : User profiles
96 : OS Registered Applications	97 : Containers
98 : WMI Software Features	99 : Packaged File Data

Table 6 Hardware Tests to be used with -excl and -incl switches to Enable and Disable specific hardware tests

Hardware Test	Hardware Test
100 : Storage Data	101 : Devices
102 : SCSI/IDE serial numbers	110 : Network data
111 : TCP/IP data	112 : IPX data
113 : Netbios Data	114 : Network Shares
120 : Bus Data	121 : PCI Cards
122 : PCMCIA Cards	123 : MCA Cards
124 : EISA Cards	125 : ISA PnP Card detection
126 : USB Data	130 : Peripherals
150: System Configuration	

Starting the Scanners

Information Collected by the Scanners

See **Help > Data Collected by the Scanners**.

Starting the Scanner Manually

DDM Inventory allows you to automatically launch your scanners using agents. We recommend that you use the agent and Enterprise Mode scanners to schedule scans regularly. However, if you need to launch them manually do the following:

Windows Scanners

The Win32 scanner comes in two versions, namely, normal and hidden.

The normal version of the Win32 scanner is a console application with a command line user interface. It shows command line output as it executes popping up a console window if one is not available.

The hidden version of the Win32 scanner is a GUI application that does not have a user interface. Since it is not a console application, it does not pop up a console window to display output. This is preferable if a completely hidden scanner is needed when executing the scanner in the manual deployment mode, for example, as part of the Windows login script. In such cases, the hidden Win32 scanner can be used. Unlike the normal Win32 scanner, it requires no console, shows no output, and executes in a completely hidden manner. By default it is called `scanwin32h-x86.exe`.



As the hidden scanner has no way to interact with the user, if an error is encountered during its operation, it will fail with the appropriate error level. The reason for failing can usually be found in the error log file.

To start the normal Win32 scanner

- 1 Locate the scanner by using the Windows Explorer.

Scanners are located in `<installDir>\DDMI\7.60\scanners\scanners`, where `<installDir>` is the DDM Inventory installation directory. By default `<installDir>` is `C:\Program Files\Hewlett-Packard`.

- 2 Double-click on the scanner icon or file name.

Alternatively, you can start the Windows scanner from a command prompt.

To start the hidden Win32 scanner

- 1 In a command prompt window, change to the directory where the scanner program is located.
- 2 Type the name of the scanner program, for example, `scanwin32h-x86`, to start the scanner.

Alternatively, you can start the Windows scanner by double-clicking on its file name in the Windows Explorer.

UNIX/Mac OS X Scanners

The methods for starting the various UNIX scanners (HP-UX, Solaris, Linux and AIX) are identical.

To start UNIX/Mac OS X scanners

- 1 Copy the scanner executable to the machine to be scanned.
- 2 Make sure that executable bit has been set (for example, for the Solaris SPARC scanner run `chmod +x scansolaris-sparc` to ensure this)
- 3 Type the name of the scanner, for example, `scansolaris-sparc`, followed by any desired scanner command line options, to run it.

You will have to type `./` in front of `scansolaris-sparc` if the current directory (`.`) is not in the `PATH` variable as shown in the following example.

```
PATH: ./scansolaris-sparc
```

The Scanning Sequence for the Scanners

A console is shown while the scanner is running. This displays the status of the scanning sequence. Any errors encountered are also shown here.

When the scan is run, the following events take place:

- Hardware scan (also contains system configuration scan)
- Software scan

Hardware Scan

Initially a copyright message is displayed, after which, hardware is detected (this too, is indicated as text mode messages).

Software Scan

The software scan commences after the hardware scan. It shows a list of directories as they are being scanned.

Scanner Error Level Codes

The scanners produce error level codes which can be used to handle situations if the scanner terminates without producing a scan file.

These error codes can, for example, be used in a batch file so that specified actions can be carried out in the event that particular error codes are returned.

These can be used to control re-scan activities when a scan has not completed successfully.

Table 7 Scanner Error Level Codes

Error Level	Description
20	Scanner terminated because it detects that it is running in a virtual environment such as a terminal services session, Virtual PC/Server virtual machine, or a non global Solaris zone.
6	Another scanner instance is already running.
5	Too Early – It is earlier than the scan days variable.
4	Fatal Error – Scanner encountered a fatal error.
3	Help Screen – Command line help screen has been requested. It is also returned if invalid command line options are specified.
2	User Abort – User aborted the scanner.
1	Exception – Scanner terminated because of an exception
0	Normal/successful exit

See [Using Error Level codes](#) on page 99 for further information.

MSI Scanner

In This Section...

- [Overview of the MSI Scanner on page 97](#)
- [Starting the MSI Scanner on page 97](#)
- [Opening the MSI Scanner Output File in the MSI Importer on page 97](#)
- [MSI Scanner Error Level Codes on page 98](#)

Overview of the MSI Scanner

The MSI scanner is a command line utility used to scan an MSI based installer, extract all required file information and write an XML file describing the installer and its contents. This XML file can then be sent to the central office where the person maintaining the application library can load it into the SAI Editor exactly as if it was the original MSI based Installer. This allows an administrator to teach an MSI installable application to the User SAI library without having to install the application.



The MSI scanner (msiscanner.exe) is not generated by the Scanner Generator. It is supplied with the software in the following location by default: C:\Program Files\Hewlett-Packard\DDMI\7.60\Common\bin

Starting the MSI Scanner

To start the MSI scanner:

- 1 From the command prompt, type the following:
`msiscanner <setup_package> <output_file>`

Where:

- <setup_package> is the path and file name of the MSI-based installer.
- <output_file> is the path and file name of the output XML file. Note that if the specified file name does not end in .xml, the MSI scanner will append an .xml extension to it.

Opening the MSI Scanner Output File in the MSI Importer

The output from the MSI scanner is usable in the MSI Importer so that you can browse the MSI and teach from it based on the XML file only.

To open the MSI scanner output file:

- 1 In the SAI Editor, select the Import MSI based Installer option from the Tools menu.
The File Open dialog box is displayed.
- 2 In the Files of Type drop-down box, select the MSI scanner output file.
- 3 Navigate to the file to be opened.
- 4 Click OK

MSI Scanner Error Level Codes

The MSI scanner produce error level codes which can be used to handle situations if the scanner terminates without producing a scan file.

These error codes can, for example, be used in a batch file so that specified actions can be carried out in the event that particular error codes are returned.

Table 8 MSI Scanner Error Level Codes

Error Level	Description
6	Unexpected error
5	Unable to open the output file
4	Insufficient space available in the Temp directory.
3	Unable to open input MSI
2	Unrecognized package
1	Incorrect parameters
0	Success

Troubleshooting

Using Error Level codes

Windows scanners produce Error Level codes that can be used to handle situations if the scanners terminate without producing a scan file.

These can be used to control re-scan activities when a scan has not completed for some reason.

Because the Error Level is available as an environment variable when the scanner finishes this can be incorporated in a log file.

For example, a Windows NT/200x/XP/Vista Scanner with ComputerName and UserName available, a simple batch file could include:

```
echo %computername%, %errorlevel%, %username% to a flag file >
%computername%.flg
```



If the scanner is terminated using the Windows Task Manager, then it is reported as successful.

Scanner Generator Errors

The most usual problems encountered when the scanner is generated result in an error message:

ERROR {value} Generating scanner

The causes can usually be identified as follows:

- The path defined for the executable scanner executable file does not exist.
- The scanner file already exists and is currently being used by another application.
- The file name chosen for the scanner executable file is invalid (that is, does not follow the MS DOS file naming conventions and/or may include illegal characters).
- Some virus protection software may prevent the Scanner Generator from creating and writing to scanner executable files.

To Resolve the Problem

- Try generating the scanner using the default settings, path and file name.
- If the previous step fails, try again selecting a file name which you are sure does not exist.

Hardware Scanning Errors

If a hardware scanning error occurs, the screen will appear to stop responding, or 'hang' during hardware scanning.

Note the hardware test which is failing. The error message is displayed in the console/shell window.

The scanner provides several command line parameter switches for disabling specific hardware detection tests. Use these command line parameter switches to disable the specific test which has failed.



Typing `-?` following the scanner file name at the command line displays a list of the available parameter switches, for example, `Scanwin32-x86 -?`

To use a parameter switch to disable a specific hardware test, enter it on the command line after the scanner file name when the scanner is started.

For example:

```
scanlinux-x86 -excl:60
```

Software Scanning Errors

The following errors may occur during the examination of files and collection of software information.

Out of Swap Space – Cannot Store More Files in Scan File

This message is displayed if there is not enough room on the hard disk drive for storing a file that has been marked for collection in the Scanner Generator.

Could Not read File <file name> - File Not Saved

This message is displayed if a file marked for collection in the Scanner Generator cannot be stored. Check to see if the file is being locked (used) by another process.

Scan File Saving Errors

The following messages may be displayed when the scanner tries to save a scan file:

Error {value} Saving Local Scan File

There may be insufficient space on the local drive that the scanner is attempting to save the scan file to. Check the available space on the local disk drive.

Another cause of this error message appearing might be that sufficient privileges do not exist to write the file or the drive cannot be accessed.

Error {value} Saving Offsite Scan File

There may be insufficient space on the offsite drive (for example the floppy disk or network drive) when the scanner is attempting to save the scan file. Check the available space on the offsite disk drive.

Additional Errors

Additional errors the user may encounter running the scanner include:

- [Not Enough Temp Space](#)
- [Compression on Netware Servers](#)
- [Slow Scanning](#)

- Virus Warning

Not Enough Temp Space

Check that the TEMP environment variable points to a valid directory with enough disk space available. If it is missing or points to an incorrect directory, set it up accordingly (for example: SET TEMP=C:\TEMP).

Compression on Netware Servers

All signatures should be off or `override.ini` must be set to ignore all files. This ensures that files are not opened and stored files are not collected. Netware compression is not dynamic such as NTFS compression in Windows NT/2000/XP/2003/Vista. Running scanners could have detrimental effects in Netware Servers if compression is being used. This is because to signature a file, the file must be decompressed and then opened by the scanner. Netware will not recompress the file, thus a capacity problem could result if the compressed volume is greater than the actual disk space available.

Slow Scanning

This may be due to real-time antivirus software being run. Any file that is opened will be checked for virus infection. Although this can be tedious, it is not advisable to disable the antivirus software for the reasons discussed in the next section.

Virus Warning

Because the scanner opens files on the computer, if there is real-time antivirus software in operation, it may detect a virus being present in a file. Depending on the virus product being used, they will have an action defined to deal with the virus. Some will try to deal with the problem and immediately disinfect the file. Others will try to move the infected file to a quarantine directory and rename its file extension.

In this case, the quarantine directory may be scanned by the scanner later during its scan.

To prevent this from happening, use the `override.ini` file with `*.vir` (where `.vir` is a typical quarantine file extension). Check the specific product to find the extension for this type of file.

Using Scanners for Manual Inventories

DDM Inventory allows you to automatically launch your scanners using agents. We recommend that you use the Windows agent and Enterprise Mode to schedule scans regularly. However, DDM Inventory scanners can be generated as stand-alone executables that can be run in a number of ways.

Once you have configured and generated the correct type of scanners for your computer population, the next issue you will face is how to execute them.

Walkround Inventory

When starting your inventory project it may be necessary to initially conduct a walkround inventory. There may be machines that are not connected to the network, or there may be a closet full of older or broken machines which may only be discovered by physically finding them.

All of these machines need to be accounted for as part of a sound asset management program. Additionally, there is user asset information such as user first name, last name and location which must initially be manually entered.

With a walkround inventory, you can execute the scanner from a floppy disk, USB memory stick or connect to a network share and run it from there.

Using a Distribution Tool

The advantage of using a distribution tool, such as HP Client Automation Enterprise is that it gives administrators the ability to deploy and execute a command on the target system as they see fit. Since this can include the deployment and execution of a DDM Inventory scanner, it allows administrators to determine at their discretion when an inventory needs to take place on the managed device.

Command Line Execution

Although the options for the scanner are normally set using the Scanner Generator, it may be necessary to change some settings to allow better operation on some machines being scanned. This may be to accommodate a 'quirky' machine or to simply change the name given to the scan file. The advantage of running a scanner from the command line is that there are numerous switches available to override options configured in the Scanner Generator. In addition, new features become available such as the option to run a scan on a scheduled basis.

For more information about command line options for the scanners, see [Command Line Parameters and Switches](#) on page 86.

8 Logging User Actions

Some users need a method of checking the DDM Inventory logs to see the actions initiated by different accounts.

The best way to find this kind of information is to go through the audit.log and discovery.log files on the DDM Inventory server. Both are available (by default) at this location:

```
C:\Documents and Settings\All Users\Application  
Data\Hewlett-Packard\DDMI\Logs.
```

By default, DDM Inventory does not log these events. If you would like to log them, you must enable the **Log User Actions** option at **Administration > System Configuration > Server configuration**.

Audit Log

Before using the audit log, make sure that the data you want to see is recorded in the audit log. The log contains information on most changes you make through the web user interface (for example, account changes and System Configuration). The resulting data appears in the following format:

```
2005-09-21 09:37:20,453 [4380] - OV::ED::Audit::log_info
C:\Perl\site\lib\OV\ED\Audit.pm (46): "admin@127.0.0.1" set
"audit_log" === to y (ConfigOption)
```

```
2005-09-21 09:41:45,040 [4380] - OV::ED::Audit::log_info
C:\Perl\site\lib\OV\ED\Audit.pm (46): "admin@127.0.0.1" set
"max_login_failure_count" === to 4 (ConfigOption)
```

```
2005-09-21 09:42:01,062 [4380] - OV::ED::Audit::log_info
C:\Perl\site\lib\OV\ED\Audit.pm (46): "admin@127.0.0.1" add
"account" "test_account"
```

```
2005-09-21 09:42:17,562 [4380] - OV::ED::Audit::log_info
C:\Perl\site\lib\OV\ED\Audit.pm (46): "admin@127.0.0.1" change
"password" of "test_account" === *****
```

For each entry, you will see the following information:

Example	Explanation
2005-09-21 09:41:45,040 [4380] -	Date and time of the change.
OV::ED::Audit::log_info C:\Perl\site\lib\OV\ED\Audit.pm (46):	The name of the script.
admin@127.0.0.1	The account name and the IP address from which the server was accessed.
set "audit_log" === to y (ConfigOption)	The UI option that was changed.

Discovery Log

The Discovery log contains events relating to discovering the network. For example, it records whenever a user updates the model of a device, or adds a new device to the range of IP addresses.

If you would like to search the log for these events, enter the following `grep` command at the DOS prompt on your DDM Inventory server.

If you have installed DDM Inventory to its default location, you can use the following command. Enter the command exactly as shown, including all punctuation marks.

```
"C:\Program
Files\Hewlett-Packard\DDMI\7.60\support\bin\grep.exe" "Event
write:" "C:\Documents and Settings\All Users\Application
Data\Hewlett-Packard\DDMI\Logs\discovery.log" | "C:\Program
Files\Hewlett-Packard\DDMI\7.60\support\bin\grep.exe" "User="
> "C:\events.txt"
```

This `grep` command filters the data in the `Discovery.log` file twice. First, it looks for data containing the text “Event write”. Second, it filters again by looking for the text “User=”.

When installing DDM Inventory 7.60, you may have changed the default location of the folders. If that is the case, make sure to change the command accordingly.

Example	Explanation
"<program files>\support\bin\grep.exe"	The location in your “program files” folder that contains the <code>grep</code> executable.
"Event write:"	The first filter for text to be found in the discovery log.
"<data directory>\Logs\discovery.log"	The location of the discovery log in your “data directory”. There are likely several <code>discovery.log</code> files in this directory, as DDM Inventory will split the file once it reaches a certain size. So, you may have to search through several log files, all following this naming convention: <code>discovery.log</code> , <code>discovery2.log</code> , <code>discovery3.log</code> .
"<program files>\support\bin\grep.exe"	The location in your “program files” folder that contains the <code>grep</code> executable.
"User=" >	The second filter for text to be found in the discovery log.
"C:\events.txt"	The name of the output file.

The resulting "events.txt" file will contain data in the following format.

```
2005-09-13 17:08:49,267 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: change Object:
node=10 Property=Title Owner=User from=<xxxx> to=<yyyy>
User="abcd" FromIP="ipv4:172.1.1.1"
```

```
2005-09-13 17:09:10,249 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: Model update mdupd
FromIP="4D58DC30" User="abcd" Type="UpdateModel" Value="Query
Device" EventTime=1126645750 Node=10
```

```
2005-09-13 17:09:17,827 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: Model update mdupd
FromIP="ipv4:172.1.1.1" User="abcd" Type="UpdateModel"
Value="Run Rulebase" EventTime=1126645757 Node=10
```

```
2005-09-13 17:09:29,763 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: Model update mdupd
FromIP="ipv4:172.1.1.1" User="abcd" Type="UpdateModel"
Value="Enrich XML" EventTime=1126645769 Node=10
```

```
2005-09-13 17:09:39,903 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: delete auto_trash
node(s) 10 User="abcd" FromIP="ipv4:172.1.1.1"
```

```
2005-09-13 17:09:45,308 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: add new device
port(s) 501 User="abcd" FromIP="ipv4:172.1.1.1"
```

```
2005-09-13 17:09:45,339 LogServerClientHandler
ovedDiscEng:Event[6236]: INFO - Event write: add node(s) 10
User="abcd" FromIP="ipv4:172.1.1.1"
```

```
2005-09-13 17:12:53,539 LogServerClientHandler
ovedDiscEng:Portal.1[9840]: INFO - Event write: Discovery
Configuration changed by User=abcd FromIP=ipv4:172.1.1.1
```

9 UI Shortcuts

You can launch major components of DDM Inventory from outside of the DDM Inventory interface. These components include the Device Manager, Port Manager, Line Manager, and all features available from the Home Page.

To launch components from outside DDM Inventory, you must use the "?go=" commands. The "?go=" commands associated with the Home Page require only a single argument. The "?go=" commands associated with the Managers can have multiple arguments.



To launch a component on a remote DDM Inventory server from a server running in Aggregator mode, use the optional argument "remote_id".

Optional arguments are shown in [square brackets]. Variables (which you must replace with a value) are shown in angle brackets and *<this font>*. You should omit the square brackets, angle brackets, and spaces between arguments when you type the actual text.

Major Components

You can launch the following major components with a single argument from a web browser:

Function	Command
Health Panel	<code>https://<my_server>/nm/?go=health_panel</code>
Network Map	<code>https://<my_server>/nm/?go=network_map</code>
Events Browser	<code>https://<my_server>/nm/?go=events</code>
Find	<code>https://<my_server>/nm/?go=find</code>
MIB Browser	<code>https://<my_server>/nm/?go=mib_browser</code>
Scan Data Viewer	<code>https://<my_server>/nm/?go=viewer</code>
Scanner Generator	<code>https://<my_server>/nm/?go=scanner_generator</code>
Home	<code>https://<my_server>/nm/?go=home</code>
Status	<code>https://<my_server>/nm/?go=status</code>
Reports	<code>https://<my_server>/nm/?go=reports</code>
Administration	<code>https://<my_server>/nm/?go=administration</code>
Help	<code>https://<my_server>/nm/?go=help</code>

Asset Questionnaire

Syntax:

`https://<my_server>/nm/?go=waq ;device=<device_id> [;device_type=<device_type>]`

Parameter	Description
device_id	any string
device_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, LabelPrefix, Label, NetBIOS

Examples:

- `https://my_server.example.com/nm/?go=waq;device=172.17.1.1`
- `https://my_server.example.com/nm/?go=waq;device=172.17.1.1;device_type=IPv4`

Device Manager

Syntax:

`https://<my_server>/nm/?go=device ;device=<device_id> [;device_type=<device_type>] [;panel=<panel>]`

Parameter	Description
device_id	any string
device_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, LabelPrefix, Label, NetBIOS
panel	one of the following options: about, state, reports, diagnosis, stats, ports, manage, update, visibility Note: If this is omitted, DDM Inventory will use the account's default setting.

Examples:

- `https://my_server.example.com/nm/?go=device;device=172.17.1.1` (open device by IP address)
- `https://my_server.example.com/nm/?go=device;device=172.17.1.1;device_type=IPv4` (open device by IP address; more efficient)

- https://my_server.example.com/nm/?go=device;device=56;device_type=NMID
(open device by internal ID)

Port Manager

Syntax:

`https://<my_server>/nm/?go=port [&device=<device_id>] [&device_type=<device_type>]
;port=<port_id> [&port_type=<port_type>] [&panel=<panel>]`

Parameter	Description
device_id	any string
device_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, Label, LabelPrefix, NetBIOS
port_id	any string
port_type	one of the following options: OID, NMID, Index, Description
panel	one of the following options: about, state, reports, diagnosis, stats, purge, connect, disconnect Note: If this is omitted, DDM Inventory will use the account's default setting.

Examples:

- https://my_server.example.com/nm/?go=port;device=172.17.1.1;port=eth0
(open port by IP address and description)
- https://my_server.example.com/nm/?go=port;port=238;port_type=NMID
(open port by internal ID)

Line Manager

Syntax:

```
https://<my_server>/nm/?go=line [&device=<device_id>] [&device_type=<device_type>]  
&port=<port_id> [&port_type=<port_type>] [&panel=<panel>]
```

Parameter	Description
device_id	any string
device_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, Label, LabelPrefix, NetBIOS
port_id	any string
port_type	one of the following options: OID, NMID, Index, Description
panel	about (the only option available)

This works the same as the Port Manager. The only additional restriction is that the Line Manager will only work if the specified port is connected to something. You may specify either end of the line.

Examples:

```
https://my_server.example.com/nm/?go=line;device=172.17.1.1;port=eth0  
(open line by IP address and description)
```

```
https://my_server.example.com/nm/?go=line;port=238;port_type=NMID  
(open line by internal ID)
```

Attribute Manager

Syntax:

```
https://<my_server>/nm/?go=attribute [;device=<device_id>] [;device_type=<device_type>]  
[;port=<port_id>] [;port_type=<port_type>] ;attribute=<attribute_id>  
[;attribute_type=<attribute_type>] [;panel=<panel>]
```

Parameter	Description
device_id	any string
device_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, Label, LabelPrefix, NetBIOS
port_id	any string
port_type	one of the following options: OID, NMID, Index, Description
attribute_id	any string
attribute_type	A full list of the internal names of these attributes is available on the web UI at Help > Classifications > Supported Device/Port Attributes.
panel	one of the following options: about, stats Note: If this is omitted, DDM Inventory will use the account's default setting.

Examples:

- `https://my_server.example.com/nm/
?go=attribute;device=172.17.1.1;port=eth0;attribute=in_util;attribute_type=Name`
(open attribute by IP address and description and show the utilization in attribute)
- `https://my_server.example.com/nm/
?go=attribute;attribute=49234;attribute_type=NMID`
(open attribute by internal ID)

Service Analyzer

Syntax:

```
https://<my_server>/nm/?go=service_analyzer [&device1=<device1_id>]
[&device1_type=<device1_type>] [&device2=<device2_id>] [&device2_type=<device2_type>]
```

Parameter	Description
device1_id	any string
device1_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, LabelPrefix, Label, NetBIOS
device2_id	any string
device2_type	one of the following options: OID, NMID, PortOID, PortNMID, IP, IPv4, IPv6, MAC, Cloud, DNS, LabelPrefix, Label, NetBIOS

Examples:

- `https://my_server.example.com/nm/?go=service_analyzer;device1=172.17.1.1;device2=nmc`
- `https://my_server.example.com/nm/?go=service_analyzer;device1=32;device1_type=NMID;device2=78;device2_type=NMID`

Index

Numerics

- 24 hours
 - bridge aging interval, 49
 - not seen, 82

A

- address types (definitions), 46
- aging interval, 49
- AIX Scanner
 - default scanner name, 84
 - scanning sequence, 95
 - software scan, 95
 - starting, 95
- Attribute Manager
 - panel elements, 54
- audit.log, 104

B

- banner
 - Device Manager, 55
- brackets around numerals, 55
- bridge aging, 49

C

- CIDR notation, 48
- CIR, redefining, 41
- Command line
 - hardware scanning errors, 99
 - how to use with scanners, 86
 - MSI scanner, 97
 - options for scanners, 87
 - starting unix scanners, 95
 - viewing options in the analysis tools, 92
- command line execution, 102
- communication models
 - FDDI, 42 to 43
 - frame relay, 40 to 41

- community strings
 - definition, 48
 - directed, 49
 - multiple strings, 48
 - SNMP traps, 48

CSV, 20

D

- definitions
 - DDM Inventory terms and concepts, 52 to 56
 - network terms and concepts, 46 to 51
- device
 - icon, 19
 - title, 52
- Device Adds/Deletes, 81
- Device Changes, 81
- Device Manager
 - banner, 55
 - panel elements, 54
 - title, 55
- directed community strings, 49
- discovery.log, 105
- Disk space
 - compression on netware servers, 101
 - not enough to run scanner, 101
- domain name
 - definition, 46

E

- Errors
 - hardware scanning, 99
 - other scanner, 100
 - scan file saving, 100
 - scanner generation, 99
 - software scanning, 100
- Exceptions, 81

F

- FDDI, 42 to 43
- frame relay, 40 to 41

G

gray background
 Manager data, 55

H

Hardware data
 scanning errors, 99

Health Panel

- Device Adds/Deletes, 81
- Device Changes, 81
- Exceptions, 81
- Not Recently Seen, 81
- Port Add/Deletes, 81
- Port Changes, 81

HP Client Automation, 102

HP-UX Scanner

- default scanner name, 84
- scanning sequence, 95
- software scan, 95
- starting, 95

HSRP, 43

I

icons

- object label, 52

IP address

- definition, 46

IPv4 address (definition), 46

IPv6 address (definition), 47

K

Keyboard

- disabling hardware detection routine, 92

L

layers 2 and 3 (OSI), 50

Linux Scanner

- default scanner name, 84
- scanning sequence, 95
- software scan, 95
- starting, 95

M

MAC address (definition), 47

- numeric, 47
- with OUI, 47

management workstation requirements, 50

manual inventories, 102
 command line execution, 102
 using HP Client Automation, 102
 workaround, 102

mask, network see netmask

MIB, 46

Mouse

- disabling hardware detection routine, 92

MSI Scanner

- browsing the MSI in MSI Importer, 97
- default scanner name, 84
- error codes, 98
- starting, 97
- what it does, 97

multiple community strings, 48

N

negative statistics, 55

netmask notation (definition), 48

Network inventory

- using HP Client Automation, 102

Not Recently Seen, 81

O

object label, 52

Organization Unique Identifier (OUI), 47

OSI model, 50

OUI, 47

P

parentheses around numerals, 55

PC requirements, 50

PC Scanners

- default scanner names, 84

Port Add/Deletes, 81

Port Changes, 81

Port Manager

- panel elements, 54

R

recorded events

- Device Adds/Deletes, 81
- Device Changes, 81
- Exceptions, 81
- Not Recently Seen, 81
- Port Add/Deletes, 81
- Port Changes, 81

S

- Saving
 - scan file errors, 100
- scan execution, 23
- scan files
 - collect and store, 23
 - enrichment, 24
- Scanner
 - default scanner name, 84
- scanner deployment
 - automatic, 23
- scanners, 83
- schedule, Network Discovery, 43
- SMT, 42
- SNMP, 46
 - traps, 48
- Solaris Scanner
 - scanning sequence, 95
 - software scan, 95
 - starting, 95
- stale data, 55
- Starting
 - MSI scanner, 97
 - UNIX scanners, 95
 - windows scanners, 94

T

- terminology definitions
 - address types, 46
 - community string, 48
 - DDM Inventory, 52 to 56
 - domain name, 46
 - IP address, 46
 - IPv4 address, 46
 - IPv6 address, 47
 - layers 2 and 3 (OSI), 50
 - MAC address (numeric), 47
 - MAC address (with OUI), 47
 - netmask notation, 48
 - network, 46 to 51
 - OSI, 50
- title
 - device, 52
 - icon, 55
 - object, 55
- title bar see banner
- traps, SNMP, 48

U

- Unix Scanners
 - default scanner names, 84
 - scanning sequence, 95
 - software scan, 95
 - starting, 95

W

- walkaround inventory, 102
- Windows 16-bit Scanner
 - error codes, 95
 - information collected, 94
 - starting, 94
- Windows 32-bit Scanner
 - default scanner name, 84
 - error codes, 95
 - information collected, 94
 - starting, 94

