HP Business Availability Center

for the Windows and Solaris operating systems

Software Version: 8.00

Discovery and Dependency Mapping



Document Release Date: January 2009 Software Release Date: January 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.

Third-Party Web Sites

HP provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. HP makes no representations or warranties whatsoever as to site content or availability.

Copyright Notices

© Copyright 2005 - 2009 Mercury Interactive (Israel) Ltd.

Trademark Notices

 $Adobe \hbox{\tt \mathbb{R} and $Acrobat$@ are trademarks of $Adobe Systems Incorporated.}$

Intel®, Pentium®, and Intel® XeonTM are trademarks of Intel Corporation in the U.S. and other countries.

JavaTM is a US trademark of Sun Microsystems, Inc.

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

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

Unix® is a registered trademark of The Open Group.

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

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

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

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

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

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

Support

You can visit the HP Software Support web site at:

http://www.hp.com/go/hpsoftwaresupport

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

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

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Table of Contents

Welcome to This Guide	11
How This Guide Is Organized	11
Who Should Read This Guide	
Getting More Information	12
ART I: THE DISCOVERY AND DEPENDENCY MAPPING PROBE	
Chapter 1: Probe Installation	15
Install the Probe	
Run Probe Manager and Probe Gateway on Separate Machines	
Configure the Probe Manager and Probe Gateway Components.	
Upgrade the Probe	
Probe Installation Requirements	
Chapter 2: The Discovery and Dependency Mapping (DDM)	
Probe	29
DDM Probe Tasks	
Data Validation on the DDM Probe	34
Filtering Results	35
Blocking the Domain Scope Document Credentials	36
Install the DDM Probe	
The DiscoveryProbe.properties File	38

PART II: ADMINISTRATION

Chapter 3: Introducing Discovery and Dependency Mapping	41
Discovery and Dependency Mapping - Overview	
Agentless Technology	
Discovery and Dependency Mapping Architecture	44
Discovery and Dependency Mapping Components	45
Discovery and Dependency Mapping Applications	48
Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs	50
Class Model Changes	52
Manually Activate a Job	52
Manually Create a Network CI	53
Schedule Modules to Run	53
Naming Conventions	53
Log Files	54
Troubleshooting and Limitations	61
Chapter 4: Run Discovery	69
Run Discovery – Overview	
Viewing Permissions While Running Jobs	
Managing Problems With Error Reporting	
Run Discovery – Basic Mode Workflow	
Run Discovery – Advanced Mode Workflow	
Manage Errors	
View Job Information on the DDM Probe	
Run Discovery User Interface	
Chapter 5: Set Up Discovery Probes	159
Job Execution Policies	159
Add a Probe	
Set Up Discovery Probes User Interface	
Domain Credential References	
Chapter 6: Manage Discovery Resources	195
Automatically Deleted System Components	196
Creating a Discovery Wizard	
Discovering Software Elements	
Configure the DDM Probe to Automatically Delete CIs –	
Workflow	199
Discover Software Elements – Scenario	
Resource Files	
Manage Discovery Resources User Interface	

	Chapter 7: Show Status Snapshot	251
	Show Status Snapshot – Overview	
	View Current Status of Discovered CIs	
	Show Status Snapshot User Interface	252
	•	
PART III:	ADVANCED DISCOVERY	
	Chapter 8: Discovery and Dependency Mapping Content	261
	Discovery and Dependency Mapping Content – Overview	
	Class Model – Overview	
	Application – Microsoft Exchange	268
	Application – SAP	
	Application – Siebel	
	Application – UDDI Registry	290
	Cluster – Microsoft Cluster	
	Cluster – Veritas	294
	Database – DB2	297
	Database – MS-SQL	299
	Database – Oracle	301
	Discovery Tools	303
	Integration – DDMi	303
	Integration – NMM Layer 2	303
	Integration – StorageEssential	303
	J2EE – WebLogic	304
	J2EE – WebSphere	305
	Network – Advanced	306
	Network – Basic	308
	Network – Credential-less	308
	Network – Host Resources	315
	Network – Layer 2	320
	Network – Load Balancer	335
	Network Connections – Active Discovery	342
	Virtualization – VMware	352
	Wah Sarvars IIS	370

	Chapter 9: Importing Data from External Sources	373
	Importing Data from External Sources – Overview	
	The External_source_import Package	376
	The Import from CSV File Job	378
	The Import from Properties File Job	380
	The Import from Database Job	381
	The External Source Mapping Files	386
	Converters	386
	Import CSV Data from an External Source – Scenario	389
	Troubleshooting and Limitations	393
	Chapter 10: Content Development and Pattern-Writing	395
	Introducing Content Development and Pattern-Writing	
	Associating Business Value with Discovery Development	
	DDM Patterns and Related Components	
	The DDM Development Cycle	
	DDM and Integration	
	Research Stage	
	Separating Patterns	
	HP Discovery and Dependency Mapping API Reference	
	Implement a Pattern	411
	Step 1: Create a Discovery and Dependency Mapping Pattern	411
	Step 2: Assign a Job to the Pattern	
	Step 3: Create Jython Code	
	Record DDM Code	
	Configure Eclipse to Run Jython Scripts in Debug Mode	
	Discovery and Dependency Mapping Code	447
	Jython Libraries and Utilities	
	Using External Java JAR Files Within Jython	
	Job and Pattern XML Formats	453
	Chapter 11: The HP Discovery and Dependency Mapping	
	Web Service API	455
	Conventions	456
	The HP Discovery and Dependency Mapping Web Service	456
	Call the Web Service	458
	Discovery and Dependency Mapping Methods	458
PART IV:	DISCOVERY AND DEPENDENCY MAPPING SECURITY	
	Chapter 12: Hardening Discovery and Dependency Mapping	
	Hardening Discovery and Dependency Mapping – Overview	
	Manage the Domain Scope Document (DSD)	470

Chapter 13: Using SSL with the Discovery and Dependency	
Mapping Probe	.475
Configure SSL from the Discovery Probe to the Gateway Server	
Index	.479

Table of Contents

Welcome to This Guide

This guide describes how to install the Discovery and Dependency Mapping (DDM) Probe, how to manage the DDM process and to automatically discover and map IT infrastructure resources and their interdependencies. DDM can discover such resources as applications, databases, network devices, different types of servers, and so on. For users with an advanced knowledge of discovery, there is a section on pattern-writing.

This chapter includes:

- ➤ How This Guide Is Organized on page 11
- ➤ Who Should Read This Guide on page 12
- ➤ Getting More Information on page 12

How This Guide Is Organized

The guide contains the following chapters:

Part I The Discovery and Dependency Mapping Probe

Explains how to install the DDM Probe and provides details on how the Probe works.

Part II Administration

Describes the main concepts, tasks, and reference for the Run Discovery, Set Up Discovery Probes, Manage Discovery Resources, and Show Status Snapshot applications.

Part III Advanced Discovery

This section is intended for users with an advanced knowledge of Discovery and Dependency Mapping. It explains how to write custom patterns and how to run discovery for many system components. This section also explains how to use the HP Discovery and Dependency Mapping Web Service API to manage discovery and dependency.

Part IV Discovery and Dependency Mapping Security

This section explains how to harden Discovery and Dependency Mapping and the DDM Probe.

Who Should Read This Guide

This guide is intended for the following users of HP Business Availability Center:

- ➤ HP Business Availability Center administrators
- ➤ HP Business Availability Center platform administrators
- ➤ HP Business Availability Center application administrators
- ➤ HP Business Availability Center data collector administrators

Readers of this guide should be knowledgeable about enterprise system administration, have familiarity with ITIL concepts, and be knowledgeable about HP Business Availability Center in general and HP Universal CMDB technology specifically.

Getting More Information

For a complete list of all online documentation included with HP Business Availability Center, additional online resources, information on acquiring documentation updates, and typographical conventions used in this guide, see the *HP Business Availability Center Deployment Guide* PDF.

Part I

The Discovery and Dependency Mapping Probe

Probe Installation

This chapter describes the procedures that are needed for the installation of the Discovery and Dependency Mapping (DDM) Probe on a Windows platform.

You cannot install the Probe on a Solaris machine. If the HP Universal CMDB server is installed on a Solaris machine, install the DDM Probes from the Windows CD-ROM.

This chapter includes:

Tasks

- ➤ Install the Probe on page 16
- ➤ Run Probe Manager and Probe Gateway on Separate Machines on page 25
- ➤ Configure the Probe Manager and Probe Gateway Components on page 25
- ➤ Upgrade the Probe on page 26

Reference

➤ Probe Installation Requirements on page 28

The Probe

The following procedure explains how to install the DDM Probe.

The Probe can be installed before or after you install the HP Business Availability Center Gateway server. However, during Probe installation you must provide the server name, so it is preferable to install the server before installing the Probe.

It is recommended to install the Probe on a separate server from the Business Availability Center servers, to distribute the overall system load.

To install the DDM Probe:

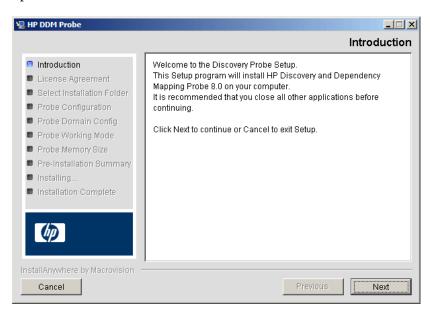
1 Select Admin > Platform > Setup and Maintenance > Downloads.

Note: The Probe link in the Downloads page is displayed only if you have purchased a standard or advanced license for Discovery and Dependency Mapping, and if the administrator has added the Probe link to the Downloads page. For details, see "Licensing Models for HP Universal CMDB" and "Installing Component Setup Files" in the *HP Business Availability Center Deployment Guide* PDF.

- **2** Click **Discovery Probe for Windows 2000/2003/XP**. You can open the Setup file or save it to your computer:
 - ➤ If you choose to open the file, it is not saved to your computer, and the setup program starts immediately. In this case, depending on your browser security settings, a security warning dialog box may open. Confirm that you want to proceed.
 - ➤ If you choose to save the file to your computer, double-click the downloaded file to begin installation.

A progress bar is displayed. Once the initial process is complete, the splash screen opens.

Choose the locale language and click **OK**. The Introduction dialog box opens.

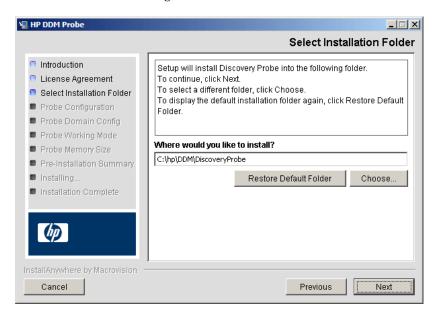


Click **Next** to open the License Agreement dialog box.



Chapter 1 • Probe Installation

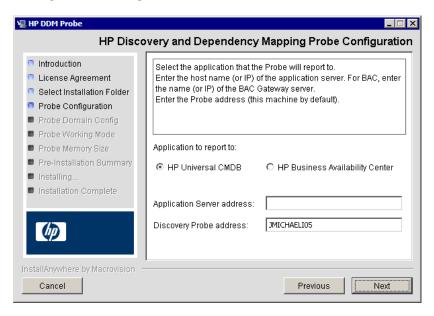
5 Accept the terms of the agreement and click **Next** to open the Select Installation Folder dialog box.



Accept the default entry or click **Choose** to display a standard Browse dialog box. To install to a different directory, browse to and select the installation folder.

Note: To restore the default installation directory, after selecting a directory in the Browse dialog box, click **Restore Default Folder**.

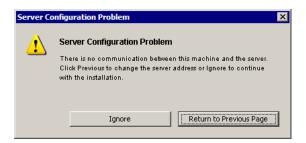
6 Click **Next** to open the HP Discovery and Dependency Mapping Probe Configuration dialog box.



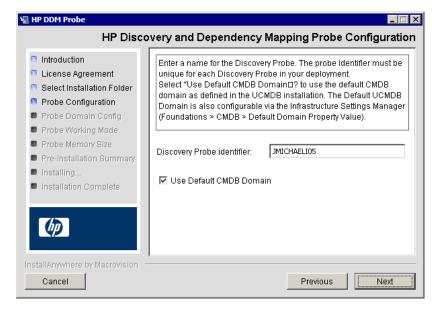
- ➤ Application to report to. Choose the application server with which you are working. You can use the Probe with either HP Universal CMDB or HP Business Availability Center.
 - ➤ If you select HP Universal CMDB, in the **Application Server address** box, enter the name or the IP address of the HP Universal CMDB server to which the Probe is to be connected.
 - ➤ If you select HP Business Availability Center, in the **Application Server address** box, enter the IP or the DNS name of the Gateway Server.
- ➤ In the **Discovery Probe address** box, enter the IP address or the DNS name of the machine on which you are currently installing the Probe, or accept the default.

Chapter 1 • Probe Installation

7 If you do not enter the address of the application server, you can choose to continue to install the Probe without the address, or to return to the previous page and add the address.



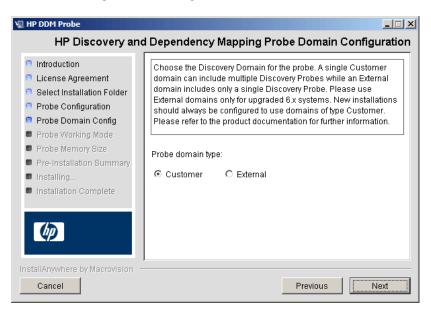
8 Click **Next** to open the HP Discovery and Dependency Mapping Probe Configuration dialog box.



➤ In the **Discovery Probe identifier** box, enter a name for the Probe that will be used to identify it in your environment.

Important: The UCMDB Probe identifier must be unique for each Probe in your deployment.

- ➤ Select **Use Default CMDB Domain** to use the default UCMDB domain as defined in the BAC Server installation. The Default UCMDB Domain is also configurable via the Infrastructure Settings Manager (**Foundations** > **CMDB** > **CMDB Class Model Settings** > **Default Domain Property Value**).
- **9** Click **Next** to open the HP Discovery and Dependency Mapping Probe Domain Configuration dialog box.



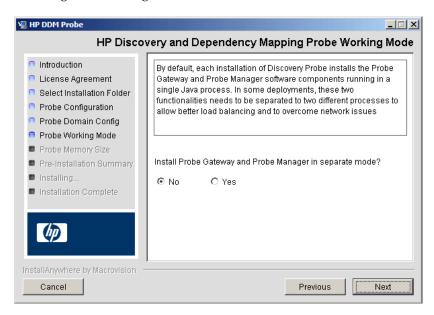
Choose between **Customer** and **External**, depending on the type of domain on which the Probe is to be running:

➤ Customer. Select if you are installing one or more Probes in your deployment.

Important: For new installations, always select **Customer**.

External. Select if you are upgrading from version 6.x systems.

10 Click **Next** to open the HP Discovery and Dependency Mapping Probe Working Mode dialog box.



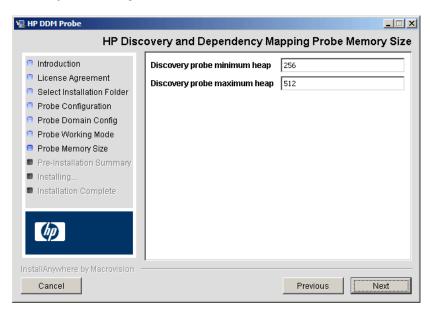
You can run the Probe Gateway and Manager as one Java process or as separate processes. You would probably run them as separate processes in deployments that need better load balancing and to overcome network issues.

Click **No** to run Probe Gateway and Probe Manager as one process.

Click **Yes** to run Probe Gateway and Probe Manager as two processes. For details on the procedure, see "Run Probe Manager and Probe Gateway on Separate Machines" on page 25.

For details on running Probe Gateway and Probe Manager, see "DDM Probe Tasks" on page 30.

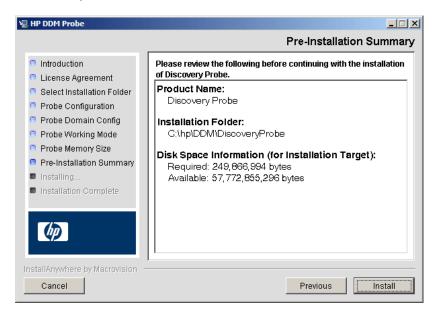
11 Click **Next** to open the HP Discovery and Dependency Mapping Probe Memory Size dialog box.



Define the minimum and maximum memory to be allocated to the Probe. The values are measured in megabytes.

Note: For large systems (more than one hundred thousand CIs), it is recommended to enlarge the minimum and maximum heaps to 1 GB. Increase your memory max heap size for the Probe Manager. Also, separate the Probe Gateway and Manager to run as individual processes (defined in the previous step).

12 Click **Next** to open the Pre-Installation Summary dialog box and review the selections you have made.



13 Click **Install** to complete the installation of the Probe. When the installation is complete the Install Complete page is displayed.

Note: Any errors occurring during installation are written to the following file: **C:\hp\DDM\DiscoveryProbe\Discovery_Probe_InstallLog.log**.

14 Click **Done**. The following shortcut is added to the Windows **Start** menu:

Programs > HP DDM > DDM Probe

15 Activate the Probe by selecting the shortcut.

The Probe is displayed in HP Universal CMDB: access Admin > Universal CMDB > Discovery > Set Up Discovery Probes > Domains and Probes. For details, see "Domains and Probes Pane" on page 173.

Run Probe Manager and Probe Gateway on Separate Machines

During installation, you can choose to separate the Probe Manager and Probe Gateway processes so that they run on separate machines. You must:

- **1** Install the Probe on both machines according to the procedure in "Install the Probe" on page 16.
- **2** Choose **Yes** in step 10 on page 22.
- **3** Perform the configuration in "Configure the Probe Manager and Probe Gateway Components" on page 25.

P Configure the Probe Manager and Probe Gateway Components

This section explains how to set up the Probe when the Probe Manager and Probe Gateway run as separate processes on two machines.

This section includes the following topics:

- ➤ "Set Up the Probe Gateway Machine" on page 25
- ➤ "Set Up the Probe Manager Machine" on page 26
- ➤ "Start the Services" on page 26

1 Set Up the Probe Gateway Machine

- a Locate the following file:C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeGateway\probe MgrList.xml.
- **b** Change:

oprobeMgr ip>=<Manager machine name in uppercase>

C Locate the following file:C:\hp\DDM\DiscoveryProbe\root\lib\collectors\DiscoveryProbe.prope rties.

d Enter the correct format:

appilog.collectors.local.ip = <GW machine name in uppercase> appilog.collectors.probe.ip = <GW machine name in uppercase>

2 Set Up the Probe Manager Machine

In DiscoveryProbe.properties change:

appilog.collectors.local.ip = <Manager machine name in uppercase> appilog.collectors.probe.ip = <Gateway machine name in uppercase>

3 Start the Services

- a On the Gateway machine start the Gateway: Start > Programs > HP DDM> DDM Gateway.
- **b** On the Manager machine start the Manager: **Start** > **Programs** > **HP DDM** > **DDM Manager**.

🦒 Upgrade the Probe

This task describes how to upgrade the DDM Probe.

1 Uninstall the Old Probe

Uninstall all existing Probes. If a Probe is running, stop it before you uninstall it.

2 Install the New Probe

For details on installation, see "Install the Probe" on page 16.

Note:

- ➤ You should install the new Probe with the same configuration, that is, use the same Probe ID, domain name, and server name as for the previous Probe installation.
- ➤ You must reactivate active jobs after the upgrade so that the newly installed Probes receive the tasks they are assigned.

Probe Installation Requirements

Hardware Requirements

Computer/processor	Windows: Pentium IV 2.4 GHz or later processor
Memory	Windows: Minimum 1 GB RAM (Recommended: 2 GB RAM)
Virtual memory (for Windows deployment)	Minimum 2 GB Note: The virtual memory size should always be at least twice the physical memory size.
Free hard disk space	Windows: Minimum 4 GB (at least 4 GB for database software and data files) (Recommended: 20 GB hard disk)
Display	Windows: Color palette setting of at least 256 colors (32,000 colors recommended).

Software Requirements

Operating system	Windows:
	 Windows 2000 Server/Advanced Server, Service Pack 4 or later Windows 2003 Standard/Enterprise editions, Service Pack 1, Service Pack 2
Java Runtime Environment	JRE 1.5.0 (installed with the product)

The Discovery and Dependency Mapping (DDM) Probe

This chapter provides information on the DDM Probe.

This chapter includes:

Concepts

- ➤ DDM Probe Tasks on page 30
- ➤ Data Validation on the DDM Probe on page 34
- ➤ Filtering Results on page 35
- ➤ Blocking the Domain Scope Document Credentials on page 36

Tasks

➤ Install the DDM Probe on page 36

Reference

➤ The DiscoveryProbe.properties File on page 38

DDM Probe Tasks

This section explains how the DDM Probe manages tasks.

The DDM Probe comprises two components: the Probe Gateway and the Probe Manager.

- ➤ The Probe Gateway provides communication (http or https) between the Probe Manager and the Business Availability Center server, for processes such as downloading tasks and returning task results.
- ➤ The Probe Manager runs the DDM process itself.

The Probe Gateway communicates with the Probe Manager using RMI.

By default, the Gateway and Manager run as a single process but they can be configured (during installation) to reside on separate processes. For details, see step 10 on page 22. Moreover, several Probe Managers can be configured to connect to a single Probe Gateway. This can be useful if a specific Probe Manager must deal with certain DDM jobs.

This section includes the following topics:

- ➤ "Stage 1. Probe Gateway" on page 30
- ➤ "Stage 2. Business Availability Center Server" on page 31
- ➤ "Stage 3. Probe Gateway" on page 31
- ➤ "Stage 4. Probe Manager" on page 32
- ➤ "Stage 5. Probe Gateway" on page 32
- ➤ "Stage 6. Probe and Server Synchronization Process" on page 33
- ➤ "Probe Configuration Update" on page 34

Stage 1. Probe Gateway

The Business Availability Center server does not initiate tasks on the Probe; it is the Probe's responsibility to request relevant tasks to run.

Stage 2. Business Availability Center Server

At the same time as the Probe requests tasks from the Gateway Server, it also sends to the server the last update time of its configuration and the last task ID received. The server returns to the Probe one of the following:

- ➤ Updated server data (in the case that the configuration on the Probe is not current). The server data includes: Python scripts, patterns, the Domain Scope Document dictionary file, and so on. For details, see "Probe Configuration Update" on page 34. For details on using the Domain Scope Document to harden DDM, see Chapter 12, "Hardening Discovery and Dependency Mapping."
- ➤ The last task sent (if there is a mismatch between the Probe and the server last task ID).
- ➤ New tasks to run (if any exist):
 - ➤ If a job has been deactivated, the server sends a **delete job** message to the Probe.
 - ➤ If a job has been activated, the server sends a **run new job** message to the Probe.
- ➤ The Business Availability Center Gateway server sends the Probe a response (in XML format) with the new task data. Each task contains the job and pattern names, and the relevant Trigger CI data.

The number of Trigger CIs for a task is limited (100 by default). For example, if an active job includes 1000 destinations, the job is sent to the Probe in 10 tasks with 100 Trigger CIs in each task.

Stage 3. Probe Gateway

- ➤ When the Probe Gateway receives tasks from the Business Availability Center server, it saves them to its local database (MySQL).
- ➤ Periodically, a thread on the Probe Gateway scans the database for tasks and sends them to the Probe Manager. This process enables load balancing in DDM when there are multiple Probe Managers for each Probe Gateway.

Stage 4. Probe Manager

- ➤ The tasks on the Probe Manager are scheduled using the Quartz third-party library. When tasks are completed, the Probe Manager sends the results (in XML format) to the Probe Gateway. (For details on Quartz, refer to the documentation at http://www.opensymphony.com/quartz/.)
- ➤ The Probe Manager receives a set of result objects. The Probe Manager first performs processing on the results (for example, filters results, runs the result redundant mechanism), and only then prepares the results for sending to the Probe Gateway.
- ➤ The results are stored in the Probe Manager database.
- ➤ A thread scans the database for results that are ready to be sent to the Probe Gateway. These results are merged into a single result, whose size does not exceed a maximum result size (currently 20,000), as defined in the discoveryProbe.properties file:

appilog.agent.local.maxTaskResultSize = 20000

When results reach the Gateway, it immediately responds with a success or failure reply. Based on this acknowledgement from the Gateway, the Probe Manager marks the results as **ack** in the database, so that they are not sent again during the next cycle.

- ➤ The task results that have been acknowledged by the Gateway remain in the Probe Manager database till they are deleted—once a week.
- ➤ When results reach the Probe Gateway, they are not sent directly to the server, but are stored in the Gateway database, to avoid flooding the server with data.

Stage 5. Probe Gateway

- ➤ A dedicated thread on the Probe Gateway scans the database and searches for task results that are ready to be sent to the server. These results are sent to the server by the Probe Gateway, using the sendResultsToServer() API.
- ➤ If the size of data that needs to be sent is too large, it is sent in chunks (max. 50,000). The information is then updated in the CMDB (using create, update, or remove).

➤ Finally, the Probe Gateway verifies that the server has finished handling the results, deletes those results from its database (so they are not sent again to the server), and continues sending results to the server, if any more results exist.

Stage 6. Probe and Server Synchronization Process

After reading a predefined number of tasks, the Probe confirms these tasks with the server. (This process prevents the need for manually reactivating patterns or jobs.)

- ➤ The Probe sends to the server the names of all activated jobs and the number of Trigger CIs for each job.
- ➤ The server checks that the number matches that in the CMDB:
 - ➤ If a job is missing from the Probe, the server redispatches the job to the Probe.
 - ➤ If the Probe has less or more than the number of CIs on the server, the server returns the names of the problematic jobs and their CIs to the Probe.
- ➤ The Probe checks the problematic jobs' list. If the Probe includes a job that does not exist on the server, the Probe sends a **remove job** remote method call to all Probe Managers.
- ➤ If the Probe includes a CI that does not exist on the server, the Probe sends a **remove CI** remote method call to all Probe Managers.
- ➤ If the server includes a Trigger CI that does not exist on the Probe, the Probe requests this CI from the server. The server returns the task (in XML format) and the Probe distributes this task.

Probe Configuration Update

- ➤ To perform discovery, the Probe needs resource data, such as the Domain Scope Document dictionary file, scripts, and so on. The Probe is updated automatically with these resources. Along with each task request from the Probe Gateway to the server, the Probe Gateway sends the last (server) update time of its latest updated resources.
- ➤ The server, before returning any new tasks, validates that there are no more recently updated resources. If there are, instead of returning the regular queued tasks for the Probe, the server returns a special, crafted task for updating the Probe's resources.
- ➤ When the Probe receives this task, it sends a **GetResouces()** request to the server, which returns a list of resources that have not been updated to the Probe. In that way the Probe is always updated with the latest system configuration files.

\lambda Data Validation on the DDM Probe

From version 7.0, the CIT model also resides on the DDM Probe. This enables data validation to take place on the Probe when receiving data from services. Problems are generated for a specific Trigger CI and displayed to the user. For details, see "Discovery Status Pane" on page 110.

The following validation takes place on the Probe:

- ➤ The CIT of the CI is compared to that in the CIT model.
- ➤ The CI is checked to verify that all key attributes are present (on condition that the CmdbObjectId attribute is not defined).
- ➤ The CI's attributes are checked to verify that they are all defined in the CIT.
- ➤ The CI's attributes of type STRING are checked to verify that they do not exceed the size limit. If an attribute is longer than the limit, DDM checks whether an AUTO_TRUNCATE qualifier is defined for the attribute. If there is a qualifier, the value is truncated and a warning message is written to the Probe error.log file.

All invalid attributes raise a CollectorsProcessException exception, which reports on a specific CI. When the Probe finds invalid data that is related to the CITs, all data that the Probe has collected on that CI is dropped by the Probe and is not sent to the server.

For details on attributes, see "CI Type Attributes" in *Model Management*.

Filtering Results

You can filter results sent by the Probe to the Business Availability Center Gateway server. You would probably need to filter irrelevant data regularly during production runs and specifically when you are testing a limited environment.

There are two levels of filtering: pattern filtering and global filtering:

- ➤ Pattern filtering. DDM filters the results for a specific pattern and sends to the CMDB only those filtered CIs. You define a pattern filter in the Results Management Pane in the Pattern Management tab. For details, see "Pattern Management Tab" on page 228.
- ➤ Global filtering. DDM filters the results of all jobs running on a Probe. You define global filters in the globalFiltering.xml file. For details, see "globalFiltering.xml" on page 206.

The order of filtering is as follows: during a run, DDM first searches for a pattern filter and applies the filter to the results of the run. If there are no pattern filters, DDM searches for a global filter and applies that filter to the results. If DDM finds no filters, all results are sent to the server.

Blocking the Domain Scope Document Credentials

The Probe's file system holds (by default) both the encryption key and the Domain Scope Document. Each time the Probe is started, the Probe retrieves the Domain Scope Document from the server and stores it on the file system. To prevent unauthorized users from obtaining these credentials, you can configure the Probe so that the Domain Scope Document is held in the Probe's memory and is not stored on the Probe file system.

To change the configuration, access **DiscoveryProbe.properties** and change:

appilog.collectors.storeDomainScopeDocument=true

to

appilog.collectors.storeDomainScopeDocument=false

The Probe Gateway and Probe Manager serverData folders no longer contain the domainScopeDocument.bin file.

For details on using the Domain Scope Document to harden DDM, see Chapter 12, "Hardening Discovery and Dependency Mapping."

🏲 Install the DDM Probe

This section describes the DDM Probe installation procedure.

Note:

- ➤ The Probe link in the Downloads page is displayed only if you have purchased a license for the DDM application.
- ➤ The managed environment is defined by the IP ranges of the domains. However, with some patterns it is possible to override this behavior and discover CIs that are out of a Probe's range.

This task includes the following steps:

- ➤ "Install the Probe" on page 37
- ➤ "Launch the Probe" on page 37
- ➤ "Run Discovery and Dependency Mapping" on page 37

1 Install the Probe

For details, see Chapter 1, "Probe Installation."

2 Launch the Probe

On the machine on which the Probe is installed, select **Start > Programs > HP DDM > DDM Probe** to start the Probe. A Command Prompt window opens. To verify that the Probe has been launched successfully, in Business Availability Center select **Admin > Universal CMDB > Discovery > Set Up Discovery Probes**. Select the Probe and, in the Details pane, verify that the status is **connected**.

For details on how the Probe works, see "DDM Probe Tasks" on page 30.

Note: To stop the Probe, in the Command Prompt window, press CTRL+C, then **y**.

Run Discovery and Dependency Mapping

For details, see "Run Discovery – Overview" on page 70.

🍳 The DiscoveryProbe.properties File

A DDM process needs several parameters to be activated. These parameters specify the method to be used (for example, ping five times before declaring a failure) and on which CI a method should be used. If parameters have not been defined by the user, the DDM process uses the default parameters defined in the **DiscoveryProbe.properties** file. To edit the parameters, open **DiscoveryProbe.properties** in a text editor.

The DiscoveryProbe.properties file is located in C:\hp\DDM\DiscoveryProbe\root\lib\collectors.

Note: If you update the **DiscoveryProbe.properties** parameters, you must restart the Probe so that it is updated with the changes.

The **DiscoveryProbe.properties** file is divided into the following sections:

- ➤ Server Connection Definitions. Contains parameters that are needed to set up the connection between the server and the Probe, such as the protocol to be used, machine names, default Probe and domain names, time-outs, and basic authentication.
- ➤ **Discovery Probe Definitions.** Contains parameters that define the Probe, such as root folder location, ports, and Manager and Gateway addresses.
- ➤ **Probe Gateway Configurations.** Contains parameters that define time intervals for retrieving data.
- ➤ **Probe Manager Configurations.** Contains parameters that define Probe Manager functionality, such as scheduled intervals, result grouping, chunking, threading, time-outs, and filtering.
- ➤ **I18N Parameters.** Contains parameters that define language settings.
- ➤ Internal Configurations. (Caution: These parameters should not be changed without an advanced knowledge of Discovery and Dependency Mapping.) Contains parameters that enable DDM to function efficiently, such as thread pool size.

Part II

Administration

Introducing Discovery and Dependency Mapping

This chapter provides information on Discovery and Dependency Mapping.

This chapter includes:

Concepts

- ➤ Discovery and Dependency Mapping Overview on page 42
- ➤ Agentless Technology on page 43
- ➤ Discovery and Dependency Mapping Architecture on page 44
- ➤ Discovery and Dependency Mapping Components on page 45
- ➤ Discovery and Dependency Mapping Applications on page 48
- ➤ Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs on page 50
- ➤ Class Model Changes on page 52

Tasks

- ➤ Manually Activate a Job on page 52
- ➤ Manually Create a Network CI on page 53
- ➤ Schedule Modules to Run on page 53

Reference

- ➤ Naming Conventions on page 53
- ➤ Log Files on page 54

Troubleshooting and Limitations on page 61

Discovery and Dependency Mapping – Overview

The Discovery and Dependency Mapping (DDM) process is the mechanism that enables you to collect information about your system by discovering the IT infrastructure resources and their interdependencies. DDM automatically discovers and maps logical application assets in Layers 2 to 7 of the Open System Interconnection (OSI) Model.

DDM discovers resources such as applications, databases, network devices, servers, and so on. DDM also communicates with industry standard or application APIs. Each discovered IT resource is delivered to, and stored in, the configuration management database (CMDB) where the resource is represented as a managed CI.

DDM is an ongoing, automatic process that continuously detects changes that occur in the IT infrastructure and updates the CMDB accordingly. You do not need to install any agents on the devices to be discovered.

Following installation, the network on which the DDM Probe is located, the host on which the Probe resides, and the host's IP address are automatically discovered and a CI is created for each of these objects. These discovered CIs are placed in the CMDB. They act as triggers that activate a DDM job. Every time a job is activated, the job discovers more CIs, which in turn are used as triggers for other jobs. This process continues until the entire IT infrastructure is discovered and mapped.

Once you configure DDM and activate the required patterns, DDM runs on the system, discovers system components, and saves them as CIs in the CMDB. You can discover new objects either manually or automatically. Objects that are outside the Probe's network require additional, manual configuration.

Note: This guide assumes that DDM Probe is installed in the default location, that is, **C:\hp\DDM\DiscoveryProbe**.

Agentless Technology

DDM is an agentless technology that discovers IT environment components through a dedicated Probe residing on the customer's site. For example, the Netlinks discovery module discovers TCP/IP connections from received NetFlow data.

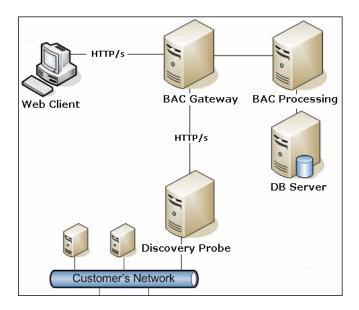
The Probe connects to HP Business Availability Center via http or https traffic to receive new tasks, send task results, and so on. For details on the Probe workflow, see "DDM Probe Tasks" on page 30.

Although DDM is agentless, that is, it does not require the installation of any agent on a customer's machine, it does depend on agents that are already installed such as:

- ➤ SNMP Agent. Provides information about the operating systems, device types, installed software, and other system resources information. SNMP agents can usually be extended to support new MIBs, exposing more data for managerial purposes.
- ➤ WMI Agent. Microsoft's remote management agent, which is usually available for access by a remote administrator. The WMI agent is also extensible by adding WMI providers to the generic agent.
- ➤ Telnet/SSH Agent (or daemon). Used mostly on UNIX systems to connect remotely to a machine and to launch various commands to obtain data.
- ➤ xCmd. A remote administration technology similar in functionality to Telnet/SSH that enables launching any console command over Windows machines. xCmd relies on Administrative Shares & Remove Service Administration APIs to function correctly.
- ➤ Application specific. This agent depends on the remote application to function as an agent and respond appropriately to the Probe's remote queries, for example, database discoveries, Web server discoveries, and SAP and Siebel discoveries.

Discovery and Dependency Mapping Architecture

Discovery and Dependency Mapping architecture is deployed as follows:



DDM User Interface

- ➤ The DDM Probe is the component that performs the data collection and runs on the customer's network.
- ➤ The user interface servlet (collectorsUtilities) and the Probe servlets (collectorsServlet, collectorsResultsServlet, and collectorsDownloadServlet) reside on the HP Business Availability Center Gateway Server machine.
- ➤ To process clients' requests, these components interact with the HP Business Availability Center Data Processing Server running the DDM components, the viewing system, and the CMDB.

Discovery and Dependency Mapping Components

This section includes the following topics:

- ➤ "DDM Probe" on page 45
- ➤ "HP Business Availability Center Servers" on page 45
- ➤ "Discovery Modules" on page 46
- ➤ "Jobs" on page 46
- ➤ "Discovery and Dependency Mapping Wizards" on page 47
- ➤ "Protocols" on page 47
- ➤ "Patterns" on page 47
- ➤ "Configuration Files" on page 47
- ➤ "External Resources" on page 48
- ➤ "Packages" on page 48
- ➤ "Scripts" on page 48

DDM Probe

The Probe is the main component responsible for requesting tasks from the server, dispatching them, and sending the results back to the CMDB through the server. You define a range of network addresses for a specific, installed Probe. Each Probe is identified by its name. For details on how the Probe functions, see "DDM Probe Tasks" on page 30.

The DiscoveryProbe.properties file contains configuration parameters. The file is located in C:\hp\DDM\DiscoveryProbe\root\lib\collectors. For details, see "The DiscoveryProbe.properties File" on page 38.

HP Business Availability Center Servers

The HP Business Availability Center Gateway Server hosts the servlets that deliver requests to the Probe. The Processing Server receives the results and stores the collected data in the CMDB.

Discovery Modules

The module is a grouping of jobs that logically belong together, can be operated and managed together, and so on. This helps to reduce clutter in the main view when many jobs need to be written, and can also offer better manageability.

When creating a job, you should choose a module for it or create a new module. If you are creating several jobs, the best practice is to split them into logical groups and assign them to modules accordingly.

Jobs

A job enables reuse of a pattern for different DDM processes. Jobs enable scheduling the same pattern differently over different sets of triggered CIs and also supplying different parameters to each set. (To activate DDM, you activate jobs—organized in modules—and not patterns.)

Jobs are organized in modules as follows:

- ➤ Network and host resources. You can discover resources on Windows and UNIX hosts, for example, disk information, running processes or services, and so on.
- ➤ Applications. The Application modules discover Oracle E-Business Suite components, the SAP environment based on Computer Center Management System (CCMS), the Siebel environment (such as the Siebel topology and database), and Web services such as the UDDI registry.
- ➤ Database modules (Oracle, DB2, Sybase, Microsoft SQL Server). DDM first finds instances of databases, then of the database resources (for example, users, tables, tablespaces) for each database instance. Business Availability Center includes predefined default views of the DB2, Oracle, and Microsoft SQL Server databases.
- ➤ **J2EE applications.** DDM discovers JBoss, Oracle Application Server, WebLogic and WebSphere components.
- ➤ Web servers. DDM discovers Apache and Microsoft IIS for Windows, SunOne for Solaris, and IBM HTTP Server.

Discovery and Dependency Mapping Wizards

You use one of the DDM wizards (to discover the infrastructure, databases, and J2EE applications) when you need to use the default values set for IP ranges, network credentials, and so on. For details on using a wizard to run DDM, see "Basic Mode Window" on page 94.

Protocols

Discovery of the IT infrastructure components uses protocols such as SNMP, WMI, JMX, Telnet, and so on. For details, see "Domain Credential References" on page 177.

Patterns

Patterns are one of the resources of a Discovery and Dependency Mapping job. A pattern includes default configuration parameters, an input TQL (that describes potential input CIs), and scheduling information that define how to perform DDM. A pattern also includes scripts and other code needed for discovery.

A job can either override the default pattern configuration (by associating a specific set of Trigger CIs with each pattern) or can run what is declared in the pattern.

For details on making pattern changes, see "Manage Discovery Resources Window" on page 224. For details on pattern-writing, see Chapter 10, "Content Development and Pattern-Writing."

Configuration Files

Configuration files include properties and parameters that are relevant for the DDM patterns. For example, the portNumberToPortName.xml file (that maps a discovered port's number to a port name) includes a list of ports used by DDM when discovering networks. For details on user-definable files, see "Resource Files" on page 204.

External Resources

External resources include all resources external to Business Availability Center that are needed in DDM, for example, a Visual Basic file, a credentials file, and so on.

Packages

Packages contain job definitions, patterns, resources, and tools that enable you to discover IT infrastructure resources such as network extensions, applications, and databases. For details, see "Package Manager" in *Model Management*.

Scripts

Business Availability Center uses Jython scripts for pattern writing. For example, the SNMP_Connection.py script is used by the SNMP_NET_Dis_Connection pattern to try and connect to machines using SNMP. Jython is a language based on Python and powered by Java. For details on pattern-writing, see Chapter 10, "Content Development and Pattern-Writing."

For details on how to work in Jython, you can refer to these Web sites:

- ➤ http://www.jython.org
- ➤ http://www.python.org

Discovery and Dependency Mapping Applications

Discovery and Dependency Mapping includes the following applications:

- ➤ "Run Discovery" on page 49
- ➤ "Set Up Discovery Probes" on page 49
- ➤ "Manage Discovery Resources" on page 49
- ➤ "Show Status Snapshot" on page 49

Run Discovery

The Run Discovery application enables you to manage the DDM modules and jobs (required for discovering a specific group of CIs). You run the process by activating jobs. You can choose to activate all or some of the jobs in a module. You can also edit jobs, and you can schedule when a job should run.

For details, see Chapter 4, "Run Discovery."

Set Up Discovery Probes

Set Up Discovery Probes enables you to add Probes to the system and to edit existing Probes. You define the network range that each Probe must cover.

For details, see Chapter 5, "Set Up Discovery Probes."

Manage Discovery Resources

Note: Only users with an advanced knowledge of Discovery and Dependency Mapping should make changes to the resources.

Manage Discovery Resources enables you to view the resources that are needed to perform discovery. You can edit patterns, scripts, configuration files, and you can replace or remove external resources needed in DDM.

For details, see Chapter 6, "Manage Discovery Resources."

Show Status Snapshot

Show Status Snapshot enables you to view details about the scheduling of a particular job as well as job statistics. You can also view the report results in a My BSM portlet.

For details, see Chapter 7, "Show Status Snapshot."

\lambda Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs

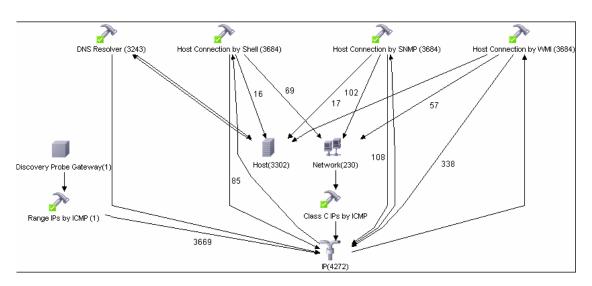
This section describes the functions of Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs. For details on these objects, see "Define Pattern Input (Trigger CIT and Input TQL)" on page 412 and other sections in Chapter 10, "Content Development and Pattern-Writing." For details on TQLs, see "Topology Query Language" in *Model Management*.

Trigger CITs

The Trigger CIT defines which CIT is used as the input for a pattern. For example, for a pattern that is going to discover IPs, the input CIT is Network.

Example – Trigger CIT Activating Jobs

The Network CIT is a trigger that activates the Class C IPs by ICMP job. The Class C IPs by ICMP job then discovers instances of IP addresses. These discovered IP addresses themselves become CIs that act as triggers to activate the Host Connection jobs, which in turn discover more IP addresses and Network CIs. The process ends when all the IP addresses included in the range defined for the Probe are discovered, as seen in the following illustration:



Trigger Cls

A Trigger CI is a CI in the CMDB that activates a job. Every time a job is activated, the job discovers more CIs, which in turn are used as triggers for other jobs. This process continues until the entire IT infrastructure is discovered and mapped.

For details on adding Trigger CIs to a job, see "Discovery Status Pane" on page 110.

Input TQLs

An Input TQL has two functions:

- ➤ An Input TQL is associated with a pattern. The Input TQL defines a minimal set of requirements for every Trigger CI included in a job that runs this pattern. (This is true even when no trigger TQL is associated with the job.)

 For example, an input TQL can query for IPs running SNMP, that is, only IPs with installed SNMP agents can trigger this pattern. This prevents the case where a user could manually create a Trigger CI that adds all hosts as triggers to a pattern.
- ➤ An Input TQL defines how to retrieve data information from the CMDB.

 Destination data information, even if it is not included in a Trigger CI, can be retrieved by the Input TQL. The Input TQL defines how to retrieve the information.

For example, you can define a relationship between a Trigger CI (a node with the node name of **SOURCE**) and the target CI and then can refer to the target CI according to this node name, in the Triggered CI Data pane. For details, see "Triggered CI Data Pane" on page 239.

For details on using input TQLs when writing patterns, see "Step 1: Create a Discovery and Dependency Mapping Pattern" on page 411.

Trigger TQLs

A Trigger TQL associated with a job is a subset of the Input TQL, and defines which specific CIs should be the Trigger CIs for a job. That is, if an Input TQL queries for IPs running SNMP, a Trigger TQL queries for IPs running SNMP in the range 195.0.0.0-195.0.0.10.

Note: A Trigger TQL must refer to the same objects as the Input TQL. For example, if an Input TQL of a pattern queries for IPs running SNMP, you cannot define a Trigger TQL for an associated job to query for IPs connected to a host. This is because some of the IPs may not be connected to an SNMP object, as required by the Input TQL.

\lambda Class Model Changes

For details on the changes to the class model in version 8.00, see "Class Model Changes" in the *HP Universal CMDB Deployment Guide* PDF. Also, see "Class Model – Overview" on page 264.

🦒 Manually Activate a Job

You can activate a job by clicking the **Activate** button in the Discovery Modules pane. You can manually activate a CI by disabling the TQL and clicking the **Add CI** button. (You disable a TQL in the **Edit Probe Limitation for TQL Output** dialog box.) The job runs using only the redispatched CIs. For details, see "Discovery Modules Pane" on page 119.

🦒 Manually Create a Network Cl

A Probe starts by discovering the network on which it is running, so usually there is no need for you to create a network CI. However, if you install the Probe on a certain network, but configure the Probe to discover objects on another network, DDM is not able to discover the network. You must manually create a network CI for the network that the Probe must discover.

To verify that a network CI exists, access the View Manager (**Admin** > **Universal CMDB** > **Modeling** > **View Manager**). Locate the Network folder and verify that the folder contains a Network Topology view.

For details on manually creating a network CI, see "New CI Dialog Box" in *Model Management*.

Schedule Modules to Run

You can set a schedule for a job or a module so that it runs at a certain time. For details, see "Discovery Scheduler Dialog Box" on page 124.

Naming Conventions

When naming entities in DDM, you can use the following characters: a-z, A-Z, 0-9. When entering IP addresses, use only digits and asterisks (*).

Log Files

This section describes the DDM log files and explains how to perform basic troubleshooting. Log files store messages, including errors, relating to the activation of jobs. For details on problem management, see "Managing Problems With Error Reporting" on page 72. For details on setting options for communication logs, see "Execution Options Pane" on page 230.

Severity Levels

Each log is set so that the information it records corresponds to a certain severity threshold. Because the various logs are used to keep track of different information, each is pre-set to an appropriate default level. For details on changing the log level, see "Changing Log Levels" below.

Severity levels are listed here from narrowest to widest scope:

- ➤ **Fatal.** This level reports serious errors such as a problem with the infrastructure, missing DLL files, or exceptions.
- ➤ **Debug.** This level is used by HP Software Support when troubleshooting problems.
- ➤ Error. This level reports problems that cause DDM not to retrieve data. Look through these errors as they usually require some action to be taken (for example, to increase time-out, to change a range, to change a parameter, to add another user credential, and so on).
- ➤ Warning. When a run is successful but there may be non-serious problems that you should be aware of, DDM marks the severity as Warning. You should look at these CIs to see whether data is missing, before beginning a more detailed debugging session. Warning can include messages about the lack of an installed agent or remote host, or that invalid data caused an attribute not to be properly calculated.
- ➤ Info. The log records all activity. Most of the information is normally routine and of little use and the log file quickly fills up.
- ➤ Success. The Trigger CI ran successfully.

Note: The names of the different log levels may vary slightly on different servers and for different procedures. For example, **Info** may be referred to as **Always logged** or **Flow**.

Changing Log Levels

If requested by HP Software Support, you may have to change the severity threshold level in a log (that is, to set verbosity), for example, to a debug level.

To change the severity threshold level:

- 1 Open the log properties file in a text editor. Log file properties are defined in files in the following directories:
 - a C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeManager\ probeMgrLog4j.properties
 - b C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeGateway\ probeGwLog4j.properties
- **2** Locate the log parameter. For example:

```
log4j.appender.LOGFILE_Performance.Threshold=DEBUG
```

3 Change the level to the required level. For example:

```
log4j.appender.LOGFILE_Performance.Threshold=INFO
```

For a description of the log levels, see "Severity Levels" on page 54.

4 Save the file.

This section includes the following topics:

- ➤ "Server Logs" on page 56
- ➤ "Probe Logs" on page 57

🍳 Server Logs

Server log files reside on the HP Business Availability Center Gateway server. They store information about server activity, including error messages, that occurs on the server side.

The following logs are located in **<HP Business Availability Center root directory>\log**.

mamAutoDiscovery.log

Description. Contains information about tasks running on the server. The server provides services to the user interface, such as: activating jobs, processing results from the Probe, or creating tasks for the Probe. In a distributed environment, the file resides on the Data Processing server.

Error Level. All DDM process errors on the server side.

Information Level. Information about requests being processed.

Debug Level. Logs mainly for debugging purposes.

Basic Troubleshooting. Check this log when you have invalid user interface responses or errors you need to explore. This log provides information to enable you to analyze the problems.

discoveryServlet.log

This log receives messages from:

- ➤ The Collectors Utilities Servlet. The user interface connects to the server through this servlet.
- ➤ The Collectors Servlet. The Probe requests new tasks from the server through this servlet.
- ➤ The Collectors Results Servlet. The Probe sends new results through this servlet.
- ➤ The Collectors Download Servlet. The Probe downloads new server data through this servlet.

In a distributed environment, the file resides on the Gateway server.

Error Level. All errors in the servlet.

Information Level. Information about user requests and Probe task requests.

Debug Level.

- ➤ User requests
- ➤ Probe requests to read DDM tasks.
- ➤ Probe access of the servlet.

Basic Troubleshooting.

- ➤ User Interface–Server communication problems.
- ➤ Probe-Server communication problems.

Some processing problems may be written to this log instead of to mamAutoDiscovery.log.

mamAutoDiscoveryUpgrade.log

Contains information about the upgrade process.

mamAutoDiscoveryResultsStat.log

Contains the statistics of the results received from the Probe.



🙎 Probe Logs

Probe logs store information involving job activation that occurs in the Probe Gateway and Probe Manager.

The logs in this section are located in C:\hp\DDM\DiscoveryProbe\root \logs.

General Logs

wrapperProbe.log

Records all the Probe's console output in a single log file.

Error Level. Any error that occurs within the Probe Gateway.

Information Level. Important information messages, such as the arrival or removal of a new task.

Debug Level. Record of every Probe access of the servlet.

Basic Troubleshooting. Use this file for any Probe Gateway problems to verify what occurred with the Probe Gateway at any time as well as any important problems it encountered.

probe-error.log

Summary of the errors from the Probe.

Error Level. All errors in the Probe components.

Information Level. N/A

Debug Level. N/A

Basic Troubleshooting. Check this log to verify if errors occurred in the Probe components.

probe-infra.log

List of all infrastructure messages.

Error Level. All infrastructure errors.

Information Level. Information about infrastructure actions.

Debug Level. Messages mainly for debug purposes.

Basic Troubleshooting. Messages from the Probe's infrastructure only.

wrapperLocal.log

Error Level. Any error that occurs within the Probe Manager.

Information Level. Important information messages such as received tasks, task activation, and the transferring of results.

Debug Level. N/A

Basic Troubleshooting. Use this file for any Probe Manager problems to verify what occurred with the Probe Manager at any time as well as any important problems it encountered.

Probe Gateway Logs

probeGW-taskResults.log

This log records all the task results sent from the Probe Gateway to the server.

Error Level. N/A

Information Level. Result details: task ID, job ID, number of CIs to delete or update.

Debug Level. The **ObjectStateHolderVecto**r results that are sent to the server (in an XML string).

Basic Troubleshooting.

- ➤ If there is a problem with the results that reach the server, check this log to see which results were sent to the server by the Probe Gateway.
- ➤ The results in this log are written only after they are sent to the server. Before that, the results can be viewed through the Probe JMX console (use the ProbeGW Results Sender MBean). You may have to log in to the JMX console with a user name and password.

probeGW-tasks.log

This log records all the tasks received by the Probe Gateway.

Error Level. N/A

Information Level. N/A

Debug Level. The task's XML.

Basic Troubleshooting.

- ➤ If the Probe Gateway tasks are not synchronized with the server tasks, check this log to determine which tasks the Probe Gateway received.
- ➤ You can view the current task's state through the JMX console (use the Discovery Scheduler MBean).

Probe Manager Logs

probeMgr-services.log

Java services debug messages.

Error Level. N/A

Information Level. N/A

Debug Level. N/A

Basic Troubleshooting. Check this log to view Java services debug messages.

probeMgr-performance.log

Performance statistics dump, collected every predefined period of time, which includes memory information and thread pool statuses.

Error Level. N/A

Information Level. N/A

Debug Level. N/A

Basic Troubleshooting.

- ➤ Check this log to investigate memory issues over time.
- ➤ The statistics are logged every 1 minute, by default.

${\bf probeMgr-patterns Debug.log}$

This log contains messages used to debug pattern issues.

Error Level. N/A

Information Level. N/A

Debug Level. N/A

Basic Troubleshooting. Use this log file for debugging patterns.

Troubleshooting and Limitations

For details on using the log files to perform basic troubleshooting, see "Log Files" on page 54.

This section includes the following topics:

- ➤ "The Probe Gateway and Probe Manager Activation" on page 62
- ➤ "The Probe Gateway and Probe Manager Connection" on page 63
- ➤ "Host Name Cannot Be Resolved to IP Address" on page 63
- ➤ "Discovery Tab Missing from Tabs" on page 64
- ➤ "Connection Fails" on page 64
- ➤ "DDM Results Do Not Appear in the Topology Map" on page 64
- ➤ "Networks and IPs" on page 64
- ➤ "TCP Ports" on page 65
- ➤ "Status 'Disconnected' for Probe" on page 65
- ➤ "SSH/Telnet Credentials" on page 65
- ➤ "SNMP Credentials" on page 66
- ➤ "Discover Resources on a Windows XP Machine" on page 66
- ➤ "SAP Discovery Fails" on page 66
- ➤ "Limitations" on page 67

The Probe Gateway and Probe Manager Activation

Problem. The Probe Gateway or Probe Manager cannot be activated.

Indication. When trying to activate the Probe Gateway or Probe Manager, the console opens and immediately closes.

Verification. To view the exception message, open the following files located in **<Business Availability Center root directory>\UCMDBServer\root\logs**:

- ➤ For the Probe Gateway: wrapperProbe.log
- ➤ For the Probe Manager: wrapperLocal.log

A message is displayed. If one of the following messages is displayed, the problem lies in the memory size definition:

Initial heap too small for new size specified. Incompatible initial and maximum heap sizes specified. The port number is being used.

To solve this problem, see the following Solution section. If another message is displayed and you cannot fix the problem, contact HP Software Support.

Solution. There can be several reasons for the activation problem, for example:

- ➤ Inappropriate memory size. Minimum and maximum memory sizes are allocated for each CMDB component. These definitions are set in the batch.cmd file, under the set memory sizes section. If memory sizes are too high for your workstation, are illegal, or incompatible with one another, you must change them, save the file, and restart the component whose values you changed.
- ➤ Installation path is too long. If you installed Business Availability Center to a directory with a long path, the operating system or JVM may have problems running the Business Availability Center execution commands. Reinstall Business Availability Center to a different directory that creates a shorter path.

The Probe Gateway and Probe Manager Connection

Problem. The connection between the Probe Gateway and Probe Manager cannot be established.

Indication. The DDM process is not working properly.

Verification. For the Probe Gateway: An error message is displayed in the Probe Gateway log (**wrapperProbe.log**, located in **<DDM Probe root directory>\DiscoveryProbe\root\logs**), as shown in the following example:

Failed to connect to probe manager at <server>. Will retry later

For the Probe Manager: An error message is displayed in the Probe Manager log (probe-infra.log, located in <DDM Probe root directory>\DiscoveryProbe\root\logs), as shown in the following example:

Connection attempt to service:jmx:rmi:///jndi/rmi://<Probe GW HOST>:1742/jmxrmi failed, probe GW may be down

Solution. Check the following:

- ➤ Verify that the correct port—1742—is defined. The RMI connection port parameter is called appilog.collectors.rmi.port. It is defined in the DiscoveryProbe.properties file, located in <Business Availability Center root directory>\UCMDBServer\root\lib\collectors.
- ➤ Verify whether the Probe Manager port is being used by another application. To verify this, in the Windows command interpreter (cmd.exe) type: netstat -na. A list of ports that are currently in use is displayed. If the port is in use, either close the other application or change the port number in the DiscoveryProbe.properties file.

Host Name Cannot Be Resolved to IP Address

Problem. A host name cannot be resolved to its IP address. If this happens, the host cannot be discovered, and patterns do not run.

Solution. Add the host machine name to the Windows HOSTS file on the Probe machine.

Discovery Tab Missing from Tabs

Problem. The Discovery tab is not displayed in the main page of Business Availability Center.

Solution. Install a license for the Probe. For details, see "Licensing Models for HP Universal CMDB" in the HP Business Availability Center Deployment Guide PDF.

Connection Fails

Problem. The connection between the HP Universal CMDB server and the Probe fails due to an RMI or http exception.

Solution. Ensure that none of the Probe ports are in use by another process.

DDM Results Do Not Appear in the Topology Map

Problem. Data that should have been discovered during the DDM process does not appear in the topology map.

Verification. The CMDB cannot retrieve the data or build the TQL results. Check the Discovery Statistics pane. If the CIs were not created, the problem is occurring during the DDM process.

Solution. Check the error messages in the **probeMgr-services.log** file located in **<DDM Probe root directory>\DiscoveryProbe\root\logs**.

Networks and IPs

Problem. Not all networks or IPs have been discovered.

Indication. Not all the networks or IPs appear in the topology map results.

Verification. The IP address range in the Set Up Discovery Probes window does not encompass the scope of the networks or IPs that should have been discovered.

Solution. Change the scope of the DDM range:

1 Select Admin > Discovery > Set Up Discovery Probes to open the Set Up Discovery Probes window.

- **2** Select the Probe and the range.
- **3** Change the IP address range in the Ranges box as required.

TCP Ports

Problem. Not all TCP ports have been discovered.

Indication. Not all TCP ports appear in the topology map results.

Verification. Open the **portNumberToPortName.xml** file (**Admin > Universal CMDB > Manage Discovery Resources > Network > Configuration Files > portNumberToPortName.xml**), and search for the missing TCP ports.

Solution. Add the port numbers that should be discovered to the **portNumberToPortName.xml** file.

Status 'Disconnected' for Probe

Problem. Discovery and Dependency Mapping shows a disconnected status for a Probe.

Solution. Check the following on the Probe machine:

- ➤ That the Probe is running.
- ➤ That there are no network problems.

SSH/Telnet Credentials

Problem. Failure to connect to the TTY (SSH/Telnet) agent.

Solution. To troubleshoot connectivity problems with the TTY (SSH/Telnet) agent, use a utility that can verify the connectivity with the TTY (SSH/Telnet) agent. An example of such a utility is the client tool PuTTY.

SNMP Credentials

Problem. Failure to collect information from SNMP devices.

- ➤ **Solution 1**. Verify that you can actually access information from your Network Management station by using a utility that can verify the connectivity with the SNMP agent. An example of such a utility is GetIf.
- ➤ **Solution 2**. Verify that the connection data to the SNMP protocol has been defined correctly in the Add Protocol Parameters dialog box. For details, see "Protocol Parameters Dialog Box" on page 175.
- ➤ **Solution 3**. Verify that you have the necessary access rights to retrieve data from the MIB objects on the SNMP agent.

Discover Resources on a Windows XP Machine

Problem. Failure to discover resources on a machine running on the Windows platform.

- ➤ Solution 1. Start > Settings > Control Panel > System. In the Remote tab, verify that the following check box is selected: Allow users to connect remotely to this computer.
- ➤ Solution 2. (For Windows XP) In Windows Explorer, select Tools > Folder Options. In the View tab, clear the Use simple file sharing (Recommended) check box.

SAP Discovery Fails

Problem. The SAP discovery fails and a Java message is displayed:

This application has failed to start because MSVCR71.dll was not found.

Solution. Two .dll files are missing. For the solution, read Note #684106 in https://websmp205.sap-ag.de/~form/sapnet?_FRAME=CONTAINER&_OBJECT=012003146900000245872003.

Limitations

- ➤ When the Discovery Probe is installed on a non-English operating system, the installation wizard remains in English.
- ➤ When DDM is installed on a non-English operating system, job and module names are limited to English characters.
- ➤ When performing an Oracle Real Application Clusters (Oracle RAC) discovery, note that DDM cannot discover links to the remote machines (the database clients) in the following situation: The discovered database reports its clients by their host names and not by their IP addresses, and the host name cannot be resolved to an IP address. In this case, the remote client cannot be created.

Chapter 3 • Introducing Discovery and Dependency Mapping

Run Discovery

This chapter provides information on using DDM to discover system components.

This chapter includes:

Concepts

- ➤ Run Discovery Overview on page 70
- ➤ Viewing Permissions While Running Jobs on page 71
- ➤ Managing Problems With Error Reporting on page 72

 Tasks
- ➤ Run Discovery Basic Mode Workflow on page 73
- ➤ Run Discovery Advanced Mode Workflow on page 74
- ➤ Manage Errors on page 78
- ➤ View Job Information on the DDM Probe on page 79

Reference

➤ Run Discovery User Interface on page 91

Run Discovery – Overview

The Run Discovery pages enable you to activate jobs that discover the components of your system. You activate DDM with one of the following methods:

➤ Use **Basic Mode** to run DDM for a specific component (for example, the infrastructure, J2EE applications, or databases), using configurable, default preferences.

For details on the workflow, see "Run Discovery – Basic Mode Workflow" on page 73.

For details on the Discovery wizard, see "Basic Mode Window" on page 94.

Note: Basic Mode is displayed by default when you access Run Discovery.

➤ Use **Advanced Mode** to run DDM to customize a run by making changes to a job.

For details on the workflow, see "Run Discovery – Advanced Mode Workflow" on page 74.

For details on the Discovery wizard, see "Advanced Mode Window" on page 93.

For details on running a specific module, see Chapter 8, "Discovery and Dependency Mapping Content."

Note: To view Help on Run Discovery components:

- ➤ For details on the Discovery Modules pane, see "Discovery Modules Pane" on page 119.
- ➤ For details on the Details tab, see "Details Tab" on page 108.
- ➤ For details on the Properties tab, see "Properties Tab" on page 146.
- ➤ For details on the Dependency Map tab, see "Dependency Map Tab" on page 106.

🖧 Viewing Permissions While Running Jobs

During a job run, you often need to know which credentials are being used to connect to a component in the system. You also often need to know the effect of a run on network performance, for example, whether the job should be run at night instead of during the day. View Permissions enables you to view the objects and parameters of a job's Jython script commands, as can be seen in the following image:

Permission	Operation	Usage Description	Objects and Parameters
shellprotocol	exec	Basic login	uname
			ver
shellprotocol	exec	CPU Info	AIX: Isattr grep "proc"
			AIX: prtconf grep "proc"
			FreeBSD: dmesg grep "cpu\ Multiprocessor"
			FreeBSD: dmesg grep -A1 "CPU:"
			FreeBSD: sysctl hw.model hw.ncpu hw.clockrate
			HPUX: model
			Linux: cat /proc/cpuinfo
			SunOS: /usr/sbin/psrinfo -v
			SunOS: prtconf
			Windows: reg query HKEY_LOCAL_MACHINE\HARDWARE\DESCRIP

Note: The information you define here is not dynamic, that is, if a pattern is changed, the information in this dialog box is not updated.

For details, see "Discovery Permissions Window" on page 123.

Example of Using the Discovery Permissions Window

You are running the Host Connection by Shell job to discover a host running on a UNIX system. An error message in the Discovery Status pane shows that DDM could not access a host through SSH because permission was denied. You display the Discovery Permissions window and see that the command to access the host requires a user with a certain level of permissions. You check the SSH Protocol window and discover that the user defined there does not have that level of permissions.

To resolve the problem, either change the user in the SSH protocol or update the permissions for the existing user in the external system.

For details on jobs that support this functionality, see "Important Information" on page 123.

\lambda Managing Problems With Error Reporting

During DDM, many errors may be uncovered, for example, connection failures, hardware problems, exceptions, time-outs, and so on. DDM displays these errors in Run Discovery, in both Basic and Advanced Mode. You can drill down from the Trigger CI that caused the problem to view the error message itself.

DDM differentiates between errors that can be ignored (for example, an unreachable host) and errors that must be dealt with (for example, credential problems or missing configuration or DLL files). Moreover, DDM reports errors once, even if the same error occurs on successive runs, and reports an error even it if occurs once only.

For details on severity levels, see "Severity Levels" on page 54.

Error Table in Database

All DDM errors are saved to the discovery_problems table in the Probe Manager database schema. (The error information is saved to the database—and is not handled in the Probe's memory—to guarantee delivery to the server.) The Probe holds the latest list of problems for each Trigger CI. After each run, the Probe checks for changes and reports them in the Discovery Status pane. For details, see "Discovery Status Pane" on page 110.

р Run Discovery – Basic Mode Workflow

This task describes how to begin mapping your system and its components, using the Discovery wizards. You run this workflow to use default values for the components in an infrastructure, database, or J2EE discovery.

Note: For details of running DDM in Advanced Mode, see "Run Discovery – Advanced Mode Workflow" on page 74.

This task includes the following steps:

- ➤ "Prerequisites" on page 73
- ➤ "Access the Discovery Wizard" on page 73

1 Prerequisites

Verify that the Probe is installed. For details on installing the Probe, see "Install the Probe" on page 16.

For details on licensing, see "Licensing Models for HP Universal CMDB" in the HP Business Availability Center Deployment Guide PDF.

2 Access the Discovery Wizard

For details, see the relevant wizard: "Infrastructure Wizard" on page 128, "J2EE Wizard" on page 136, or "Database Wizard" on page 99.

Run Discovery – Advanced Mode Workflow

This task describes how to begin mapping your system and its components. You would use this workflow to customize the components of a module.

Note: For details of running discovery in Basic Mode, see "Run Discovery – Basic Mode Workflow" on page 73.

This task includes the following steps:

- ➤ "Prerequisites" on page 74
- ➤ "Determine Network Range" on page 74
- ➤ "Set Relevant Credentials" on page 75
- ➤ "Activate Relevant Jobs" on page 75
- ➤ "Make Changes to Relevant Patterns" on page 76
- ➤ "Monitor the DDM Process" on page 76
- ➤ "View Result Statistics" on page 77
- ➤ "Troubleshoot the Results" on page 78

1 Prerequisites

a Verify that the Probe is installed. For details on installing the Probe, see "Install the Probe" on page 16.

For details on licensing, see "Licensing Models for HP Universal CMDB" in the HP Business Availability Center Deployment Guide PDF.

b Verify that the packages are deployed.

For details, see Chapter 17, "Package Manager."

2 Determine Network Range

You must define the network range of the network to be discovered. For details, see "Add/Edit IP Range Dialog Box" on page 164.

Note: Patterns try to connect to every IP in a range. Therefore, if a range is wide, network performance may be affected.

3 Set Relevant Credentials

To enable DDM to connect to servers or applications using specific protocols, you must set the relevant credentials (for example, NTCmd, SNMP, TTY, or WMI). For details on protocol parameters, see "Domain Credential References" on page 177. For details on the Details pane in the Set Up Discovery Probes window, see "Details Pane" on page 169.

Note: DDM tries to connect to a host by using each credential in turn. DDM then saves the successful credential. The next time DDM connects to this host, it first tries to connect using the successful credential.

4 Activate Relevant Jobs

Once you have defined the network range and set credentials, you can run discovery on specific jobs. For details, see Chapter 8, "Discovery and Dependency Mapping Content."

Tip: You can view a full description of a job in the Run Discovery Properties tab, under the DDM pattern name.

Example – Finding SNMP Connections

You can search for all jobs that discover SNMP connections: in the **Run Discovery > Discovery Modules** pane, click the **Search Discovery Job** icon. In the **Find Job**s dialog box, enter **SNMP** in the **Name** box and click **Find All**. For details, see **Add Rule** in "Discovery Modules Pane" on page 119 and "Find Jobs Dialog Box" on page 127.

5 Make Changes to Relevant Patterns

You can customize patterns to discover infrequent system components. For details on pattern writing, see Chapter 10, "Content Development and Pattern-Writing."

Important: Do not make changes to default patterns without consulting HP Software Support.

6 Monitor the DDM Process

For details on monitoring the CIs that are discovered by the run, see "Statistics Results Pane" on page 117.

a Define a TQL

You create a TQL query that retrieves information about CIs and CITs from the CMDB. For details, see "Define a TQL Query" in *Model Management*.

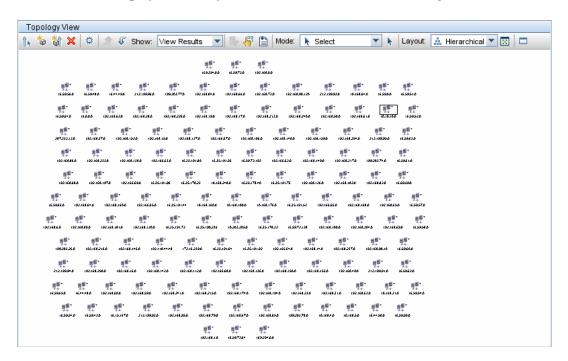
If necessary you can trigger TQLs to manually discover objects. For details, see "Trigger TQLs Pane" on page 150.

b Build a View for each TQL

A view enables you to build a subset of the overall IT universe model, containing only those CIs in the CMDB that relate to a specific discovery. For details, see "View Manager Window" in *Model Management*.

Example - Creating a View to Display Discovered CI Instances

To view the number of instances found by Business Availability Center, select **Admin > Universal CMDB > Modeling > IT Universe Manager**, and display the view you created, as seen in the following illustration:



7 View Result Statistics

You can display overall statistics for a job or you can filter the results by time range or by Probe. Each time you log in to Business Availability Center and access Run Discovery, the statistical data is updated so that the data displayed is the latest for the selected module or job.

For details on working with the statistical data, see "Statistics Results Pane" on page 117.

You can view discovered CIs also by accessing the Show Status Snapshot pane. For details, see Chapter 7, "Show Status Snapshot."

8 Troubleshoot the Results

You can check DDM results to see which errors are being reported. For details, see "Manage Errors" on page 78.

🦒 Manage Errors

This task describes how to investigate problems that arise during a run.

Note: For details about severity levels and so on, see "Managing Problems With Error Reporting" on page 72.

This task includes the following steps:

- ➤ "Prerequisites" on page 78
- ➤ "Run the Discovery Wizard or Select the Job" on page 78
- ➤ "Locate the Problem CI" on page 78
- ➤ "Troubleshoot the Problem" on page 79

1 Prerequisites

Set up DDM. For details, see "Run Discovery – Basic Mode Workflow" on page 73 or "Run Discovery – Advanced Mode Workflow" on page 74.

2 Run the Discovery Wizard or Select the Job

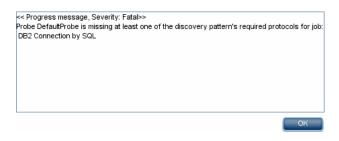
In Basic Mode, you can view error messages for a default job. In Advanced Mode, you can view error messages for one job, one module, or all modules. For details on running a wizard in Basic Mode, see "Run Discovery – Basic Mode Workflow" on page 73. For details on running a job, see "Run Discovery – Advanced Mode Workflow" on page 74.

3 Locate the Problem CI

Use the Discovery Status pane to drill down to the error messages. For details, see "Discovery Status Pane" on page 110.

Example

DDM displays the error message:



4 Troubleshoot the Problem

- ➤ For Fatal errors, you should contact HP Software Support.
- ➤ For other errors, check the CIs. For example, a Trigger CI that does not fall within the Probe's range may show an error.
- ➤ For details on setting communication logs, see "Execution Options Pane" on page 230.
- ➤ For details on managing problems, see "Managing Problems With Error Reporting" on page 72.

🦒 View Job Information on the DDM Probe

This task describes how to invoke job information (for example, job threads and trigger CIs) saved to the DDM Probe's MySql database. You work with the JMX console.

This task includes the following steps:

- ➤ "Access the MBean Operations" on page 80
- ➤ "Locate the Operation to Invoke" on page 80
- ➤ "Run the Operation" on page 91

1 Access the MBean Operations

Use the following procedure to access the JMX application on the DDM Probe and to invoke the JMX operations.

a Launch the Web browser and enter the following address:

http://<machine name or IP address>.<domain name>:1977/

where **<machine** name or **IP** address> is the machine on which the DDM Probe is installed. You may have to log in with the user name and password.

b Click the Local_<machine name or IP address> > type=JobsInformation link.

2 Locate the Operation to Invoke

In the MBean View page, locate the operation:

activateJob

Enter the name of a job to activate it immediately. This operation returns a message, for example, <job name> was triggered.

Note: The following message is displayed if the job has not been activated and there is no information about the job in the Probe's database:

Job '<job name>' does not exist in the Jobs Execution table (job was not activated!).

${\bf activate Job On Destination}$

Enter the name of a job and a Trigger CI to activate the job immediately on a specific Trigger CI. This operation returns a message, for example, The operation returned with the value: Job <job name> was triggered on destination <CI name>.

Note: Both the **JobID** and **triggerCI** fields are mandatory.

start/stop

These operations start and stop the JobsInformation service. Do not use these operations; instead, restart the Probe itself.

viewJobErrorsSummary

Enter the name of a job to return a list of error messages reported on this job, together with the error severity, the last time that the error was reported, and the number of Trigger CIs that have the error.

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

Click the entry in the **Number of trigger Cls** column to view a list of one job's trigger Cls with errors in the **viewJobTriggeredClsWithErrorId** page.

viewJobExecHistory

Enter the name of a job to retrieve a history of job invocations. A message is displayed showing the job invocations (the last invocation is shown first).

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

For each invocation the number of Triggered CIs and the total running time is shown. The Execution Details column shows at which times the job was executed. If the Probe shut down in the middle of a job execution and then resumed running or if there were blackout periods during the job execution, several time ranges are shown.

viewJobProblems

Enter the name of a job or the name of a trigger CI to retrieve a list of Trigger CIs that have problems.

Note: You must fill in at least one of the fields.

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

viewJobResultCiInstances

Fill in one or more of the parameters to return a list of CIs that have been discovered by a job.

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

The Object State Holder column displays the code for the CI or relationship defined in the CMDB. For details on creating object state holders for common CITs, see **modeling.py** in "Jython Libraries and Utilities" on page 449. For details on the ObjectStateHolder method, see the *HP Discovery and Dependency Mapping API Reference*.

viewJobResults

Fill in one or more of the parameters to return a list of CIs that have been discovered by a job.

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

When **Hide Touched Cls Info** is set to **True**, the results page displays the following information:

Column	Description
СІ Туре	Click to filter the list to show results for one CIT only.
Total CIs	Click to go to the viewJobResultCiInstances page, to view a list of all CIs that have been discovered by a job.
Triggered CIs	Click to go to the viewJobTriggeredCIs page, to view a list of all Trigger CIs that have been discovered by a job.
Last Discover Time	The date and time that the job was invoked.

When **Hide Touched CIs Info** is set to **False**, the results page displays the following information:

Column	Description
СІ Туре	Click to filter the list to show results for one CIT only.
Touched CIs	Click to go to the viewJobResultCiInstances page, to view a list of those CIs discovered by the job that are Touched CIs . For details, see
Non Touched CIs	Click to go to the viewJobResultCiInstances page, to view a list of those CIs discovered by the job that are not Touched CIs.
Triggered CIs for Touched CIs	Click to go to the viewJobTriggeredCIs page, to view a list of those Trigger CIs included in a job that are Touched CIs.
Triggered CIs for Non Touched CIs	Click to go to the viewJobTriggeredCIs page, to view a list of those Trigger CIs included in the job that are not Touched CIs.
Last Discover Time	The date and time that the job was invoked.

You can further filter the results by using the table to enter text filters.

viewJobsStatuses

Click the **viewJobsStatuses** button to return status and scheduling information for all jobs. You can choose to filter the results. For details, see "Job Operation Parameters" on page 89.

The results page displays the following information:

Column	Description
No.	The number of the job in the list.
Job Name	The job name as it appears in DDM, for example:
	Host Resources by SIMP Host Resources by SIMP Host Resources by WMI LEF JBoss Click a job to go to its viewJobStatus page, to view its status and scheduling information.
Status	The severity of the job's status, as calculated by the Probe.
	Blocked. Not in use.
	Removed. The job is no longer active.
	Running. The job is currently running.
	Scheduled. The job is scheduled to run. For details on scheduling jobs, see "Discovery Scheduler Dialog Box" on page 124.
	A red background signifies that a thread has run longer than expected and may be stuck. A green background signifies that the job is running as expected.
Errors	The number of errors for a specific job. Click to go to the viewJobErrorsSummary page, to view a list of error messages reported on this job.
Triggered CIs	The Trigger CIs that have been run by the job. Click to go to the viewJobTriggeredCIs page.
Last Invocation	The date and time that the job was last run.

Column	Description
Next Invocation	The date and time that the job is next scheduled to run.
Last Total run duration (seconds)	The total time that it took for the job to run in the last invocation. Compare this result with the average time taken for a job to run. The discrepancy is probably due to periods of time when a job waits for another job to finish.
Avg run duration (seconds)	The average time that the job ran, calculated from all previous invocations.
Recurrence	The number of times that the job was invoked. Click to go to the viewJobExecHistory page, to retrieve a history of job invocations.
Results	The number of CITs that have been discovered by the job. Click to go to the viewJobResults page to view the CITs.

viewJobStatus

Enter the name of a job to return its status and scheduling information.

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

The results page displays the following information:

Column	Description
Threading info	The total number of worker threads created by the invocation, the free worker threads, and the stuck worker threads.
Total work time	The time that the Probe took to run this job.
Tasks waiting for execution	A list of jobs together with the number of Trigger CIs that are awaiting activation.
Max. threads	The number of threads that are serving this job.

Chapter 4 • Run Discovery

Column	Description
Progress	A summary of the current run, that is, since the specific run was activated.
	For example, Progress: 2017 / 6851 destinations (29%) means that out of 6851 CIs, 2017 CIs have already run.
Working Threads information	Thread Name. The thread that is now running this job. Click to go to the viewJobThreadDump page. You use this page when a thread is running for a long time, and you must verify that this is because the thread is working hard, and not because there is a problem.
	Curr Dest. ID. The name of the host on which the job is running.
	Curr Dest. IP. The IP for which the job is discovering information.
	Work Time (Sec). The length of time that this thread is running.
	Communication Log. Click to go to the viewCommunicationLog page, to view an XML file that logs the connection between the Probe and a remote machine. For details, see "Create communication logs" on page 230.

Column	Description
Discovery Jobs Information table	Status. The severity of the job's status, as calculated by the Probe. For details, see "Status" on page 84.
	Errors. Click to go to viewJobErrorsSummary page, to view a list of error messages reported on this job.
	Triggered Cls. Click to go to viewJobTriggeredCls page, to view a list of Trigger Cls that are part of a job.
	Last invocation. The date and time that the job was last run.
	Next invocation . The date and time that the job is next scheduled to run.
	Last Total run duration (seconds). For details, see "Last Total run duration (seconds)" on page 85.
	Avg run duration (seconds) . For details, see "Avg run duration (seconds)" on page 85.
	Recurrence . The number of times that the job was invoked. Click to go to viewJobExecHistory page, to view a history of job invocations.
Results	The number of CITs that have been discovered by the job. Click to go to the viewJobResults page to view the CITs.

${\bf view Job Triggered Cls}$

Fill in one or more of the parameters to return a list of Trigger CIs that are part of a job.

For details on the job operation parameters, see "Job Operation Parameters" on page 89.

The results page displays the following information:

Column	Description
No.	The number of the job in the list.
Triggered CI ID	The CI instances that have been discovered by the job. Click to go to the viewJobResults page to view information about their CITs.
Last Execution	The date and time that the job was last run.
Service Exec. Duration (ms)	The maximum time that it took for a job to run in the last invocation, including periods when the job did not run. Compare this result with the total execution duration.
	For example, when several jobs run simultaneously, but there is only one CPU, a job might have to wait for another job to finish. The service duration includes this waiting time, whereas the total duration does not.
Total Exec. Duration (ms)	The time that it took for a job to run in the last invocation, excluding the periods when the job did not run.
Last Run Status	The status of the last run, that is, whether the run succeeded or failed. In case of failure, click to go to the viewJobProblems page, to view a list of Trigger CIs with problems.
hostID	The name of the machine on which the job is running.
ip_address	The IP address for which the job is collecting information.
	Note: If the Probe has not discovered the ID of the host (for example, because the job is given a range to discover and not a specific IP), information on the host may be missing. In this case, the hostID, ip_address, and ip_domain fields display the string (Empty).
ip_domain	The Probe name as it appears in DDM.

viewJobTriggeredClsWithErrorld

Note: This operation is part of the inner interface and serves as a helper function. Do not use this page to view Trigger CIs information; instead, use the **viewJobTriggeredCIs** page.

Job Operation Parameters

The following list includes job operation parameters.

- ➤ **ciType**. The name of the CI type (for example, ip, host).
- ➤ data. A textual field in the DiscoveryResults table that contains information about the discovered object. For example:

```
<object class="ip">
<attribute name="ip_probename" type="String">EBRUTER02</attribute>
<attribute name="ip_address" type="String">16.59.58.200</attribute>
<attribute name="ip_domain" type="String">DefaultDomain</attribute>
</object>
```

- ➤ Error Id. The error message hash string (error hash ID) that is displayed in the Jobs_Problems table.
- ➤ **HideRemovedJobs.** True: do not display jobs that have run previously and are not relevant to the current run.
- ➤ **Hide Touched CIs Info**. Touched CIs are CIs which were discovered in previous invocations. DDM already has information about these CIs, so there is no need for the Probe to send the information to the server again. The server identifies that these CIs are relevant and that there is no need to enforce the aging mechanism on them.

True: the table displays the total number of CIs and the total number of Trigger CIs for each CIT. **False:** The table displays the total number of CIs and Trigger CIs divided between touched CIs and non-touched CIs.

Chapter 4 • Run Discovery

➤ includeNonTouched. Enables filtering the table to view non-touched CIs. Choose between viewing non-touched CIs only, all CIs (touched and non-touched), or none:

	Non-touched CIs	All CIs	No CIs
(boolean)includeTouchedCis	O True ⊙ False	⊙ True ○ False	O True ⊙ False
(boolean)includeNonTouchedCis			O True ⊙ False

- ➤ includeNonTouchedCls. See includeNonTouched.
- ➤ includeTouched. Enables filtering the table to view touched CIs. Choose between viewing touched CIs only, all CIs (touched and non-touched), or none.
- ➤ includeTouchedCls. See includeTouched.
- ➤ **jobID**. The name of the job, for example, **Host Resources by SNMP**:



- ➤ maxRows. The maximum number of rows that should be displayed in the results table. The default is 100 or 1000.
- ➤ maxTriggeredCls. See maxRows.
- ➤ **objectID**. The CMDB object ID.
- ➤ showRemovedJobs. Shows information about jobs that are not currently scheduled to run, but that have run previously. These jobs take the state of REMOVED.
- ➤ showResults. Indicates whether to display the Show Results column. If the Show Results column is present, you can navigate from viewJobsStatuses to viewJobResults.
- ➤ **triggerCI**. The CMDB object ID of the trigger for a job.
- ➤ triggeredCilD. See triggerCl.

3 Run the Operation

➤ Click the button to run the operation. A message is displayed with the results of the operation run.

Reload. The number of seconds between automatic reloads of the JMX interface. **0**: The interface is never reloaded. Click the **Reload** button to manually reload the current page (if more operations have been added or removed).

Unregister. Do not touch (the view becomes inaccessible to the application that is running).

Run Discovery User Interface

This section describes:

- ➤ Advanced Mode Window on page 93
- ➤ Basic Mode Window on page 94
- ➤ Choose CIs to Add Dialog Box on page 96
- ➤ Choose Discovery TQL Dialog Box on page 97
- ➤ Choose Probe Dialog Box on page 98
- ➤ Configuration Item Properties Dialog Box on page 98
- ➤ Create New Discovery Job Window on page 98
- ➤ Database Wizard on page 99
- ➤ Dependency Map Tab on page 106
- ➤ Details Tab on page 108
- ➤ Discovered CIs Dialog Box on page 119
- ➤ Discovery Modules Pane on page 119
- ➤ Discovery Permissions Window on page 123
- ➤ Discovery Scheduler Dialog Box on page 124
- ➤ Edit Probe Limitation for TQL Output Dialog Box on page 126

Chapter 4 • Run Discovery

- ➤ Edit Time Template Dialog Box on page 126
- ➤ Find Jobs Dialog Box on page 127
- ➤ Infrastructure Wizard on page 128
- ➤ J2EE Wizard on page 136
- ➤ Properties Tab on page 146
- ➤ Related CIs Window on page 152
- ➤ Show Results for Triggered CI Page on page 152
- ➤ Source CIs Dialog Box on page 153
- ➤ Time Templates Dialog Box on page 153
- ➤ Triggered CIs on page 154
- ➤ Trigger TQL Editor Window on page 154

Advanced Mode Window

Description	Enables you to view and manage modules and jobs, to activate jobs, and to follow job progress.
	Advanced mode includes the following panes:
	➤ Discovery Modules pane. Each module includes jobs. You activate a module or job to discover a specific group of CIs. For details, see "Discovery Modules Pane" on page 119. Note: Basic Mode is displayed by default when
	accessing Run Discovery.
	➤ Details tab. Enables you to manage a module's CIs and view CI statistics. For details, see "Details Tab" on page 108.
	➤ Properties tab. Enables you to view and administer the properties of modules and jobs. For details, see "Properties Tab" on page 146.
	➤ Dependency Map. Displays a visual representation of the real-time progress of the process. For details, see "Dependency Map Tab" on page 106.
	To access: Admin > Universal CMDB > Discovery > Run Discovery.
Important Information	Each change you make in Run Discovery is delivered to and stored in the CMDB. From there, the changes are sent to the Probe. You can verify that changes have been sent to the Probe by opening the wrapperProbe.log file located in C:\hp\DDM\DiscoveryProbe\root\logs\ and searching for the following lines:
	processing document domainScopeDocument.bin
	Processing document domainScopeDocument.bin is done.
	Note : Basic Mode is displayed by default when accessing Run Discovery.
Included in Tasks	"Run Discovery – Advanced Mode Workflow" on page 74

8 Basic Mode Window

Description	Enables you to use a Discovery wizard to discover networks, databases, and J2EE applications.
	Basic mode includes the following panes:
	➤ List of wizards. Enables you to choose the wizard to run. For details, see "Infrastructure Wizard" on page 128, "Database Wizard" on page 99, or "J2EE Wizard" on page 136.
	➤ Summary pane. Enables you to run the wizard and to stop DDM running. For details, see "Summary Pane" on page 95.
	➤ Discovery Progress pane. Enables you to view a brief run status and to drill down to problematic Trigger CIs. This pane is displayed once discovery has been run for a component. For details on managing errors, see "Discovery Status Pane" on page 110.
	To access: Admin > Universal CMDB > Discovery > Run Discovery
Important Information	Note : Basic Mode is displayed by default when accessing Run Discovery.
	For details on Advanced Mode, see "Advanced Mode Window" on page 93.
Included in Tasks	"Run Discovery – Basic Mode Workflow" on page 73
Useful Links	"Run Discovery – Overview" on page 70

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A-Z)	Description
S	Click to refresh the list of wizards.

GUI Element (A-Z)	Description
<page-header> Basic Mode</page-header>	(Currently displayed) Click to run DDM for a specific component (for example, the infrastructure, J2EE applications, or databases), using configurable, default preferences.
♠ Advanced Mode	Click to run DDM when you need to customize a run by making changes to a job, pattern, and so on. For details, see "Advanced Mode Window" on page 93.

Summary Pane

Description	Enables you to run a Discovery wizard. To access: Admin > Universal CMDB > Discovery > Run Discovery
Important Information	Depending on whether a wizard has already run, the Summary pane displays the following information:
	➤ If a wizard has not yet run, the Summary pane displays the steps to be performed in the wizard and the Configure and Run button.
	➤ If a wizard has run, the Summary pane displays a summary of the run parameters, the Configure and Stop Discovery buttons, and the results of the previous run in the Discovery Progress pane.
	To run a discovery, select a wizard in the left pane and click Configure or Configure and Run to open the Discovery wizard.
	To stop a discovery run, click Stop Discovery .
Included in Tasks	"Run Discovery – Basic Mode Workflow" on page 73

Choose Cls to Add Dialog Box

Description	Enables you to choose CIs to run with selected jobs.
	To access:
	➤ Discovery > Run Discovery. In the Details tab, locate the Discovery Status pane. Click the Add CI button.
	➤ In the Oracle TNSName File Location page of the Database Wizard, click the Add CI button.

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Add CI button	Note: If you choose CIs with an error status to add to the trigger list, a message is displayed when you click the Add button.
Search Cls	Contains filters with which you can limit the number of CIs that appear in the Search Results pane.
	➤ By Discovery TQL . Select a Discovery TQL to search for those CIs that match the TQL.
	➤ Show only Cls containing. To search for CIs that include a certain text, enter the text here.
	➤ Exact match. Select to search for CIs with the exact match of the text label. (By default, you search by entering part of a text. For example, searching for 10 within the IP CIs finds all the IPs that contain 10 in their address. Entering 10 then selecting Exact match finds no results.) ➤ Search Click to display the search results
	➤ Search . Click to display the search results.

GUI Element (A–Z)	Description
Search Results	Displays a list of triggered CIs answering to the criteria set in the filter. To add the CIs to the list in the triggered CIs pane, select the CIs. You can make multiple selections.
	➤ CIT. The CI type of the selected triggered CI.
	➤ CI. The label of the triggered CI.
	➤ Related Host . The label for the host related to the triggered CI.
	➤ Related IPs. The IPs of the related host.
	Page. The list of CIs is divided into pages. The number in the Page box indicates which page is currently displayed. To view other pages, use the up and down arrows, or type the page number, and press Enter .
	To determine the number of CIs that appear on a page, right-click either the up or down button and choose the required number. The default is 25.

Q Choose Discovery TQL Dialog Box

Description	Enables you to add a trigger TQL to a job.
	To access: Click the Add TQL button in the Trigger TQLs
	pane.

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
<discovery name="" tql=""></discovery>	The TQLs that can query the CMDB for the selected CIT.
TQL Preview	Hold the cursor over an element to view details.

Q Choose Probe Dialog Box

Description	Enables you to filter the Probe list.
	To access: Click a Filter button in the Run Discovery > Details tab:
	➤ Triggered Cls pane Filter button. For details on the menu options, see "Discovery Status Pane" on page 110.
	➤ Statistics pane Filter button. For details on the menu options, see "Statistics Results Pane" on page 117.

Q Configuration Item Properties Dialog Box

Description	Enables you to view CI properties.
	To access: In the Discovered CIs dialog box, right-click a CI and choose Properties .
Important Information	For details, see "Configuration Item Properties Dialog Box" in <i>Model Management</i> .

Create New Discovery Job Window

Description	Enables you to create a job.
	To access: Right-click a module in the Discovery Modules pane, and choose Create New Job .
Important Information	For details on the panes in this window, see: * "Discovery Job Details Pane" on page 109 * "Parameters Pane" on page 149 * "Trigger TQLs Pane" on page 150 * "Global Configuration Files Pane" on page 237 * "Discovery Scheduler Pane" on page 146

Database Wizard

Description	Enables you to discover databases such as DB2, Oracle, Microsoft SQL, and Sybase. To access: Admin > Universal CMDB > Discovery > Run Discovery > Basic Mode. Select the Database wizard from the list in the left pane. Click Configure and Run.
Important Information	For more information, hold the pointer over a question mark icon: Preferences
	Choose the configuration options to be used during Discovery. IP Discovery Strategy ? Send ping request to every Send ping request only to d Network Topology (Layer 2) ? Host TCP Connections ? Activate to discover DNS nameservers and t only if zone transfer can be performed from
Wizard Map	The Database Discovery wizard contains: Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

Q Define Credentials

Description	Enables you to configure connection data for each protocol.
Important Information	You configure protocols depending on what must be discovered and which protocols are supported on your site's network.
	For a list of protocols, see "Domain Credential References" on page 177.
	General information about the wizard is available in "Database Wizard" on page 99.
Wizard Map	The Database Discovery wizard contains: Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
+	Add new connection details for selected protocol type.
×	Remove a protocol.
	Edit a protocol. Click to open the Protocol Parameters dialog box.

GUI Element (A–Z)	Description
₩ ₩	Move a protocol up or down. DDM executes all the protocols in the list with the first protocol taking priority.
Protocol	Click to view details on the protocol, including user credentials. Note: A missing credential is represented by the following icon: Domains and Probes Domains and Probes DefaultDomain DefaultDomain DefaultDomain DesaultDomain DesaultDomai
	—≣ 9 WebLogic Protocol

Database Port Scanning

Description	Enables you to discover the port itself and subsequently to discover the database.
Important Information	General information about the wizard is available in "Database Wizard" on page 99.
Wizard Map	The Database Discovery wizard contains: Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

Chapter 4 • Run Discovery

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
+	Click to add a port to the port list. The Add New Port dialog box opens. Select the ports and click OK.
	To edit existing system ports, in the Add New Port dialog box, click Edit Known System Ports . The Edit Known System Ports dialog box opens. Select the port and click the Edit button. In the dialog box that opens, make changes to the entries and click OK .
	To add a port to the list, in the Edit Known System Ports dialog box, click the Add button. Enter details of the port name, number and type and click OK .
×	Select a port and click the button to remove the port from the list.

Custom JDBC Drivers

Description	Enables you to select the JAR file for the DB2 and Sybase JDBC drivers.
Important Information	General information about the wizard is available in "Database Wizard" on page 99.
Wizard Map	The Database Discovery wizard contains: Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
DB2 JDBC Driver	Select the check box and click Import file to locate the appropriate JAR file in the DB2 JDBC installation, as follows:
	➤ db2java.zip ➤ db2jcc.jar
	▶ db2jcc_license_cu.jar▶ db2jcc_license_cisuz.jar
Sybase JDBC Driver	Select the check box and click Import file to locate the jconnectXXX.jar JAR file in the Sybase JDBC installation.

💐 Oracle TNSName File Location

Description	Enables the discovery of Oracle databases. You provide the location of the TNSNames.ora configuration file that contains database information needed to discover Oracle databases, such as port, host, SID, and so on.
Important Information	General information about the wizard is available in "Database Wizard" on page 99.
Wizard Map	The Database Discovery wizard contains: Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

Chapter 4 • Run Discovery

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Server Host	Select the hosts on which the TNSNames.ora file is located. Click the Add CI button to choose the Trigger CIs that represent these hosts. For details, see "Choose CIs to Add Dialog Box" on page 96.
	 CIT. The CI type of the selected triggered CI. CI. The label of the triggered CI. Related Host. The label for the host related to the trigger CI. Related IPs. The IPs of the related host.
TNSNames.ora file location	Enter the location of the TNSNames.ora file in the server host system. You can enter several locations (separate the locations by commas). If you terminate the path with a delimiter (for example, c:\temp\), DDM assumes that the file name is tnsnames.ora .

Schedule Discovery

Description	Enables you to define a schedule for a specific job.
Important Information	General information about the wizard is available in "Database Wizard" on page 99.
Wizard Map	The Database Discovery wizard contains:
	Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
%	You define a time template in the Discovery Scheduler pane of the Properties tab. For details, see "Discovery Scheduler Pane" on page 146.
Allow Discovery to run at	Choose the time at which the schedule should run.
Repeat Every	Select how often the schedule should run.

Summary

Description	Enables you to review the wizard definitions before running a discovery.
Important Information	To make changes to the run, click the Back button. General information about the wizard is available in "Database Wizard" on page 99.
Wizard Map	The Database Discovery wizard contains: Database Wizard > Define Credentials > Database Port Scanning > Custom JDBC Drivers > Oracle TNSName File Location > Schedule Discovery > Summary

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Run	Click the button to run a discovery.

Q Dependency Map Tab

Description	Displays a visual representation of the real-time progress of the discovery process. The map displays: CIs that were triggered by a job CIs that were discovered as a result of the activated job. To access: Click the Dependency Map tab in the Run Discovery window.
Important Information	Depending which level you select in the Discovery Modules pane, different information is displayed in the Dependency Map tab. If you select: The Discovery Modules root, and select the Show only active Discovery jobs check box, the Dependency Map displays only active jobs and their interdependencies. The Discovery Modules root, and clear the Show only active Discovery jobs check box, the Dependency Map displays all Discovery jobs and their interdependencies. A module, a topology map is displayed showing the module's active and inactive jobs. A job, the topology map highlights the job in the module's map.
Useful Links	"Discovered CIs Dialog Box" on page 119

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
<right-click menu=""></right-click>	Use the right-click menu to view details for a job, CI, or link, for example, the number of CI instances (of a specific type) in the CMDB or the number of CI instances created by a specific job.
	Depending on which object is selected, the following menu options are displayed:
	➤ When a job is selected:
	Show discovered Cls. Click to view the CIs discovered by the job. To filter the query, select a CIT from the menu.
	Show trigger Cls. Click to view the CIs that triggered the job.
	➤ When a CI is selected:
	Show all CIT instances. Click to view all CIs of this CI type.
	➤ When a link from a CI to a job is selected:
	Show trigger CIs for job. Click to view CIs (of the selected type) that triggered the job.
	➤ When a link from a job to a CI is selected:
	Show discovered instances. Click to view CIs (of the selected type) that were discovered by the job.
<toolbar></toolbar>	For details, see "Toolbar Options" in Model Management.
<tooltip></tooltip>	Hold the pointer over a CI or job to display a description.
Show only active Discovery jobs	When the Discovery Modules root is selected in the Discovery Modules pane, this check box is displayed.
	Select to display all active jobs (from any module).

Q Details Tab

Description	Enables you to view and administer modules and jobs, to follow the progress of the DDM process, and to manage errors during discovery. To access: Click the Details tab in Run Discovery.
Important Information	Depending which level you select in the Discovery Modules pane, different information is displayed in the Details tab. If you select:
	➤ The Discovery Modules root or a Discovery module, the Discovery Status and Statistics Results panes are displayed with information and statistics about all active jobs and errors discovered during a run. For details, see "Discovery Status Pane" on page 110 and "Statistics Results Pane" on page 117.
	➤ A job, the Discovery Job Details, Discovery Status, and Statistics Results panes are displayed. For details, see "Discovery Job Details Pane" on page 109, "Discovery Status Pane" on page 110, and "Statistics Results Pane" on page 117.
	➤ Several jobs or modules, the Selected Items pane is displayed. For details, see "Selected Items Pane" on page 116.
Included in Tasks	"Managing Problems With Error Reporting" on page 72

Discovery Job Details Pane

GUI Element (A–Z)	Description
	Click to go to the pattern in the Discovery Resources window.
Niew Cls in Map	You can choose to view a map of the CIs and links that are discovered by the pattern, instead of a list. Click the button to open the Discovered Classes Map window. The selected pattern is shown together with its CIs and relationships. Hold the cursor over a CIT to read a description in a tooltip.
F View Permissions	Click to view permissions that are defined for specific patterns. For details, see "Discovery Permissions Window" on page 123 For details on editing these permissions, see "Permission Editor Dialog Box" on page 241.
Discovery Pattern	The pattern needed by the job to discover the CIs.
Discovered CIs	The CIs that are discovered by this job.
Input CI Type	The CIT that triggers the CIs for this job.
Job Name	The name and description of the job and the package in which it is located.

Discovery Status Pane

Description	Enables you to view a run status and to drill down to problematic Trigger CIs, to uncover specific problems that DDM encountered during the run, for example, incorrect credentials. In Basic Mode, enables you to view the results of the previous run for the selected job type (that is, infrastructure, database, or J2EE application). In Advanced Mode, enables you to view the results of the
	previous run for a selected module or job, or for all modules. To access:
	➤ In Basic Mode, locate the Discovery Overview pane.
	➤ In Advanced Mode, select a module or job, click the Details tab, and locate the Discovery Status pane.
Important Information	➤ You can use the SHIFT and CTRL keys to select adjacent and non-adjacent CIs in a list.
	➤ Depending which level you select in Advanced Mode in the Discovery Modules pane, information is displayed in the Discovery Status pane for all modules, for a specific module, or for a specific job.
	➤ The information in this pane is automatically refreshed every thirty seconds.
Included in Tasks	"Manage Errors" on page 78
	"Check Status of Application Discovery (Rediscover a View)" in <i>Model Management</i> .
Useful Links	"Managing Problems With Error Reporting" on page 72

GUI Element (A-Z)	Description
1	Click to return to the upper pane.
F	Click to drill down to the Trigger CI that includes the problem.
	Note : This icon is displayed only when you can drill down from error or warning links.
0	Click to refresh the status view.
÷	Click to add a newly-discovered CI.
©	Click to rerun the discovery.
×	Click to remove a CI from the list, if the CI is no longer of interest. The CI is deleted from the specific job.

GUI Element (A-Z)	Description
7	Click and choose an option from the menu:
<u>u</u>	➤ By Status. Displays a list of Trigger CIs:
	➤ All. Displays all the Trigger CIs.
	➤ Waiting for Probe. Displays the Trigger CIs that are ready to be dispatched and are waiting for the Probe to retrieve them.
	➤ In Progress. Displays the Trigger CIs that are active and are running on the Probe.
	➤ In progress (being removed). Displays the Trigger CIs that are being removed from the Trigger CIs list.
	➤ Success, Failed, Warning. Displays only those CIs that have the selected status.
	➤ Discovery Errors. Displays CIs with an error status.
	➤ By Probe . Displays only the CIs triggered by a selected Probe. Click to open the Choose Probe dialog box.
	➤ By Dispatch Type. Displays a list of CIs according to one of the following options:
	➤ All. Displays both CIs that are used to manually activate the job and Discovery TQLs that are used to automatically activate the job.
	➤ Manually added. Displays the CIs that are used to manually activate the job.
	 ➤ By Discovery TQL. Displays the CIs that are used to automatically activate the job. ➤ Reset. Click to remove any filters.
	Click to display a message box containing an explanation of the failure. (You can also view messages by right-clicking the CI and selecting Show error details .)
	Click to open the Triggered CIs dialog box with additional information about the CI. For details, see "Triggered CIs" on page 154.
	Show statistics for a specific Trigger CI.
Q	Find a CI.

GUI Element (A-Z)	Description
0	Click to run the job.
<drill down=""></drill>	You can drill down from a job or a module.
	➤ Drill down from a job to view a list of Trigger CIs that are included in the job.
	➤ Drill down from a module to view a list of the jobs in the module and the number of CIs returned by each job. Drill down from a job to its Trigger CIs. Note: A Trigger CI can be present in more than one job.

GUI Element (A-Z)	Description
<right-click ci="" menu=""></right-click>	Right-click a CI to:
	 Show error details. Opens the Message dialog box with details of all errors for the selected Trigger CIs. Remove. Select to delete the CI from the job. The CI is removed from that job only, even if it appears in more than one job.
	➤ Run now. To run a specific CI or set of CIs, select the CIs. They are added to the list of CIs that the Probe is going to run (Waiting for Probe).
	➤ Show results for triggered CIs. DDM sends an ad-hoc request to the Probe and retrieves the latest results of the job (CIT name and number of discovered CIs) that is running on a specific trigger CI.
	This ad-hoc request does not run the job, but brings the results of the previous job run that are stored in the Probe's database. If the job has not yet run for this trigger CI, a message is displayed.
	For details, see "Show Results for Triggered CI Page" on page 152.
	If no communication log exists in the Probe, a message is displayed. You can choose that DDM always creates communication logs. For details, see "Execution Options Pane" on page 230.
	➤ Debug . Choose between:
	➤ View communication log for triggered CI. Opens the log that includes information about the connection between the Probe and the remote machine. This is on condition that you have set the Create communication log to either Always or On failure. For details, see "Execution Options Pane" on page 230.
	➤ Go to pattern . Displays the pattern that is included in the job in Manage Discovery Resources.
	➤ Go to job . Displays the job in which the CI is included.
	➤ Undispatch. Removes the Trigger CI.

GUI Element (A-Z)	Description
Failed	Displays those CIs that returned an Error or Fatal severity.
	Show errors. Displays a list of the different types of errors returned by these CIs. For details on severity, see "Severity Levels" on page 54.
In progress	Displays the number of Trigger CIs that are awaiting their turn to be run. Click to view the jobs that are waiting to be run. Double-click a job to view the CIs that are waiting to be run.
Look for	To search for a specific Probe, related host, or related IP, enter part of its name in the box and click Search .
Progress	The indicator shows a summary of the current discovery run, that is, since the specific run was activated.
Success	DDM displays the number of CIs that have been run successfully, that is, without errors.
	Click to view the jobs (and the number of CIs in each job) that completed successfully. Double-click a job to view information about all its CIs.
	Select a CI and use the <right-click ci="" menu=""> to view information</right-click>
	With warnings. Click to view a warning message for each job.
	Double-click a message to view the CIs that finished successfully with a warning.
	Right-click a message to view the <right-click ci="" menu="">.</right-click>
Total	Displays the status of all of a job's Trigger CIs. Double click a Warning or Error status to open the Message dialog box.
Waiting for Probe	The Trigger CIs that are either waiting for the Probe or are waiting to run.

Selected Items Pane

GUI Element (A–Z)	Description
<right-click menu=""></right-click>	Edit Scheduling. Click to open the Discovery Scheduler to define a schedule for a specific job. For details, see "Discovery Scheduler Pane" on page 146.
invoke immediately	 ➤ A check mark signifies that the Discovery job runs as soon as the triggered CI reaches the Probe. In this case, the Invoke on new triggered CIs immediately check box is selected in the Properties tab. ➤ If this column does not contain a check mark, the job runs according to the schedule defined in the Schedule Manager.
Job name	The name of the job.
Schedule info	The scheduling information of the job as defined in the Discovery Scheduler.
Trigger TQLs	The name of the TQL that activated the job. For details, see "Trigger TQLs Pane" on page 150.

Statistics Results Pane

GUI Element (A–Z)	Description
	Select a CI and click the View Instances button to view CI instances and their attributes. The Discovered CIs dialog box opens.
	Under the following conditions, a message is displayed:
	➤ All the CIs that were discovered by this job were already discovered by another job.
	➤ All the CIs that this job discovered have been deleted.
	➤ The CI instances were discovered in a previous version. (In version 7.0, you cannot view instances of CIs discovered in a previous version.)
	Note:
	➤ You can also view CI instances by double-clicking a row.
	➤ CITs with no instantiated instances are displayed.
V	Select the time range or Probe for which to display statistics about the CITs.
	➤ By Time Range:
	➤ All. Displays statistics for all job runs.
	➤ Last Hour/Day/Week/Month. Choose a period of time for which to display statistics about the CITs.
	➤ Custom Range. Click to open the Customize Statistics Time Range dialog box. Enter the date or click the arrow to choose a date and time from the calendar, for the To and From dates. To delete a date, click Reset.
	➤ By Probe : To view statistics for a specific Probe, select to open the Choose Probe dialog box.

GUI Element (A–Z)	Description
S	Click to retrieve the latest data from the server (job results are not automatically updated in the Statistics pane).
<column title=""></column>	Click a column title to change the order of the CITs from ascending to descending order, or vice versa.
<right-click a="" title=""></right-click>	Choose from the following options:
	 Hide Column. Select to hide a specific column. Show All Columns. Displayed when a column is hidden. Customize. Select to display or hide columns and to change the order of the columns in the table. Opens the Columns dialog box.
	 Auto-resize Column. Select to change a column width to fit the contents. For details, see "Select Columns Dialog Box" in Reference Information.
CIT	The name of the discovered CIT.
Created	The number of CIT instances created in the period selected or for the selected Probe.
Deleted	The number of CIT instances deleted in the period selected or for the selected Probe.
Discovered CIs	The number of CIs that were discovered for each CI type.
Filter	The time range set with the Set Time Range button.
Last updated	The date and time that the statistics table was last updated for a particular job.
Total	The total number of CIs in each column.
Updated	The number of CIT instances that were updated in the period selected.

Q Discovered CIs Dialog Box

Description	Enables you to view CI instances of a CIT discovered by a job. To access:
	 In the Statistics Results pane, select a CIT, and click the View instances button. In the Dependency Map tab, select Show Discovered CIs or Show all instances.
Important Information	 The Discovered by <job name=""> window includes the same information as the Element Instances window. For details, see "Element Instances Dialog Box" in Model Management.</job> Depending on whether you select Show discovered CIs or Show all instances in the Dependency Map, you can view either all CIs discovered by a selected job or all CIs of a selected type.

Discovery Modules Pane

Description	Enables you to view and manage modules and jobs. Each module includes the jobs necessary to discover specific CIs.
	To access: Admin > Universal CMDB > Discovery > Run Discovery. The default view is called Basic Mode and displays the Discovery Wizard. You can run the J2EE, database, or infrastructure discovery. Click Advanced Mode to view all modules.
Important Information	Caution: Only administrators with an expert knowledge of the DDM process should delete modules.
	Obsolete. Contains several modules that are no longer relevant but remain for backward compatibility and upgrade purposes. Do not use these modules on new installations.
	No module . Contains jobs that are not included in any other module.

Chapter 4 • Run Discovery

GUI Element (A–Z)	Description
6	Refresh All. Updates the modules.
Q	Find Job. Click to open the Find Jobs dialog box. For example, to search for all jobs that discover SNMP connections, click the Filter icon. In the Find Jobs dialog box, enter SNMP in the Name box and click Find All . For details, see "Find Jobs Dialog Box" on page 127.
Activate	Click to run a module or job. You can choose to activate all jobs in a module or some of them:
	 To activate all jobs in a module, right-click the module and select Activate. To activate specific jobs in a module, right-click the job
	and select Activate .
	Select the jobs or modules to be stopped and click Deactivate .
Alle	Represents the module root.
-	To create a module, right-click to enter the name of the module you are creating.
	Note: A name is case sensitive. Names beginning with an upper case letter appear in the Discovery Modules list before names beginning with a lower case letter.
=	Represents a module. Click the key to display the jobs in a module.
7	Represents a job. Click to display information about the job. To view a pattern description, hold the pointer over a job.
	Jobs contain configuration information derived from patterns and other resources and are the entities controlled by users, for example, when activating or deactivating a module.
	For details on the right-click menu, see "Right-Click Menu" on page 122.

GUI Element (A–Z)	Description
15	One green dot signifies that some of a module's jobs are activated:
	Database - Oracle TNS Oracle Config Files by Shell Oracle Credentials by SQL Oracle TNSName by Shell
	Three green dots signify that all of a module's jobs are activated:
	Database - Oracle TNS Soracle Config Files by Shell Oracle Credentials by SQL Oracle TNSName by Shell
:	An exclamation mark signifies that one or more of the jobs is experiencing a problem that could affect the DDM process, for example, a protocol connection failure.
	To view the reason for the problem, click the Handle errors button in the Triggered CI pane. For details, see "Discovery Status Pane" on page 110.
	Note : If a problem is resolved by clicking the Refresh All button, the Problem Indicator disappears.
备 Basic Mode	Click to run DDM for a specific component (for example, the infrastructure, J2EE applications, or databases), using configurable, default preferences. For details, see "Basic Mode Window" on page 94.
Advanced Mode	(Currently displayed) Click to run DDM to customize a run by making changes to a job, pattern, and so on.

Right-Click Menu

GUI Element (A–Z)	Description
Activate	Click a module to run all its jobs. To run a specific job, select and activate it.
	The Discovery Module discovers CITs and relationships of the types that are described in each job, and places them in the CMDB. For example, the Class C IPs by ICMP job discovers the Depend, IP, and Member CITs and relationships.
Create New Job	Click to open the Create New Discovery Job. For details, see "Create New Discovery Job Window" on page 98.
Create New Module	Click to define a new name for the module root.
Deactivate	Stop the module or job from running.
Delete	Click and answer Yes to the warning message.
Delete job	Click and answer Yes to the warning message.
Edit Pattern	Click to edit the pattern in the Manage Discovery Resources window.
Edit Scheduling	Click to open the Discovery Scheduler to define a schedule for a specific job.
Rename job	Click to open the Choose Name dialog box. Enter a new name for the job.
	Note: You cannot rename active jobs.
Run Now	Click to run the job again using the selected Trigger CIs.
Save as	Click to clone the job.

Q Discovery Permissions Window

Description	To access: Run Discovery > Advanced Mode. Select a job. Locate the Discovery Job Details pane in the Details tab. Click the View Permissions button.
Important Information	In version 8.00, you can view permissions data for the following jobs:
	➤ Collect Network Data by Shell or SNMP
	➤ Host Connection by Shell
	➤ Host Connection by SNMP
	➤ Host Connection by WMI
	➤ Host Resources by Shell
	➤ IIS Applications by NTCMD
	➤ MQ by Shell
	➤ MS Cluster by NTCMD
	To obtain a list of other jobs that include this feature, contact HP Software Support.
Useful Links	➤ "Viewing Permissions While Running Jobs" on page 71
	➤ "Required Permissions Pane" on page 238
	➤ "Permission Editor Dialog Box" on page 241

GUI Element (A-Z)	Description
€ Export •	Export a permission object in Excel, PDF, RFT, CSV, or XML format. For details, see "Browse Mode" in <i>Model Management</i> .
Objects and Parameters	The commands that appear in the relevant Jython scripts.
Operation	The action that is being run.
Permission	The name of the protocol as defined for the job.
Usage Description	A description of how the protocol is used.

Q Discovery Scheduler Dialog Box

Description	Enables you to define a schedule for a specific job, for example, every day DDM starts running an IP ping sweep on class C networks at 6:00 AM. To access: ➤ Right-click a job and choose Edit scheduling. ➤ Click the Edit Scheduler button in the Discovery Scheduler pane of the Properties tab in the Run Discovery window.
Important Information	The Discovery Scheduler defines the frequency of the discovery (daily, monthly) whereas the time template defines when the job should run (during the day, at night, at weekends only). You can run the same schedule with different time templates. For example, you can define a schedule that runs every day and you can define a time template that runs at night from 01:00 AM to 05:00 AM. A job defined in this way runs every day from 01:00 AM to 05:00 AM. You can define a second time template to run at a different time, and you can use this time template too with the same schedule. For details on creating a time template, see "Edit Time Template Dialog Box" on page 126.

GUI Element (A–Z)	Description
Validate Expression	Click to validate the Cron expression you entered.
<frequency></frequency>	➤ Once. Define the task to run only once.
	➤ Interval. Defines the interval between successive runs.
	➤ Daily. Run a task on a daily basis.
	➤ Weekly. Run a task on a weekly basis.
	➤ Monthly. Run a task on a monthly basis.
	➤ Cron. Enter a Cron expression in the correct format.

GUI Element (A–Z)	Description
Days of month	(Displayed when you select Monthly) Click the button to choose the days of the month on which the action must run. The Select Days dialog box opens. Choose the required days by selecting the check boxes. You can select multiple days. ➤ Select all. Select all the days. ➤ Unselect all. Clear all the selected days.
Days of the week	(Displayed when you select Weekly) Select the day or days on which the action should run.
End by	Select the date and time when the action should stop running by selecting the End by check box, opening the calendar, selecting the date and time, and clicking OK .
	Note : This step is optional. If you do not need to specify an ending date, leave the End by check box cleared.
Invocation Hour	(Displayed when you select Daily , Weekly , or Monthly) Select the time to activate the action. Click the button to open the Select Hours dialog box. Choose the required time by selecting the check boxes. You can select multiple times.
	➤ Select all. Select all the times.
	➤ Unselect all. Clear all the selected times.
	Note: You can also enter the time manually in the Invocation hour box. Separate times by a comma and enter AM or PM after the hour. The manually entered action times are not restricted to the hour and half hour only: you can assign any hour and minute combination. Use the following format: HH:MM AM, for example, 8:15 AM, 11:59 PM.
Invocation Time	(Displayed when you select Once) Choose the date and time the action should begin running by opening the calendar and choosing a date and time, or accept the default.
Months of the year	(Displayed when you select Monthly). Select the month or months in which the action must run.
Repeat every	(Displayed when you select Interval) Type a value for the interval between successive runs and choose the required unit of time (minutes, hours, or days).

GUI Element (A–Z)	Description
Start at	Choose the date and time when the action must begin running by selecting the Start at check box, opening the calendar, selecting the date and time, and clicking OK .
Time Zone	Select the time zone according to which the Probe must schedule jobs.
	The default is << Discovery Probe Time Zone >>: the Probe uses its own system-defined time zone. This enables scheduling to take place at different times in different geographical locations.
	For all Probes to start working at the same time, select a specific time zone. (This assumes that the Probes' system date and time and time zone are correctly configured.)

Lesson : Edit Probe Limitation for TQL Output Dialog Box

Description	Enables you to change the Probes on which a trigger TQL is running. For details on selecting the Probes, see "Selecting Probes" on page 177.
	To access: Select a job and click the following button: Run Discovery > Properties tab > Trigger TQLs pane > Probe Limit box.

Lesson : Edit Time Template Dialog Box

Description	Enables you to define a time template to use when scheduling jobs.
	To access: Click the Add button in the Time Templates dialog box.
Important Information	The name of the time template must be unique.
Useful Links	"Discovery Scheduler Dialog Box" on page 124

GUI Element (A–Z)	Description
Every day between	Define a daily schedule when a job must run. You can also type in times. You can assign any hour and minute combination.
Time Template	Enter a unique name.
Week Time	Define a weekly schedule when a job must run. Click to select a time. To select adjacent cells, click and drag the pointer over the table. To clear a time, click a cell a second time.

🖎 Find Jobs Dialog Box

Description	Enables you to search for jobs answering to specific criteria.
	The results of the search are displayed in the Selected Items pane in the Details tab.
	To access: Click the Filter button in the Discovery Modules pane.

GUI Element (A–Z)	Description
Direction	Searches forwards or backwards through the modules.
Find All	All jobs meeting the search criteria are highlighted.

GUI Element (A–Z)	Description
Find Discovery job by	Choose between:
	➤ Name. Enter the name, or part of it, of the job.
	➤ Input type. CIs that triggered the job. Click the button to open the Choose Configuration Item Type dialog box. Locate the CI type that you are searching for.
	➤ Output type. CIs that are discovered as a result of the activated job.
Find Next	The next job meeting the search criteria is highlighted.

1 Infrastructure Wizard

Description	Enables you to run discovery on the networks in your system.
	To access: Admin > Universal CMDB > Discovery > Run Discovery > Basic Mode. Select Infrastructure Wizard from the list in the left pane. Click Configure and Run.
Wizard Map	The Infrastructure Discovery wizard contains:
	Infrastructure Wizard > Define IP Ranges > Define Credentials > Preferences > Schedule Discovery > Summary.

Q Define IP Ranges

Description	Enables you to set the network range for discovery for
	each Probe. The results are retrieved from the addresses
	in the range you define. You can also define IP addresses
	that must be excluded from a range.

Important Information	Any changes made here affect the global configuration. General information about the wizard is available in "Infrastructure Wizard" on page 128.
Wizard Map	The Infrastructure Discovery wizard contains: Infrastructure Wizard > Define IP Ranges > Define Credentials > Preferences > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
+	For details, see "Add/Edit IP Range Dialog Box" on page 164.
×	Select a range and click the button to remove the range from the list.
	Select a range and click the button to edit an existing range.
Address Ranges	 Range. For details on the rules for defining ranges, see "Range" on page 166. Excluded. You can exclude part of a range. Select the range and click the Add button. In the dialog box, click the Advanced button. For details, see "Exclude Ranges" on page 165.
Discovery Probes	Enables you to view details on the Probe, including range information. You can also add ranges to, or exclude ranges from, the Probe. For details on defining a Probe, see "Domains and Probes Pane" on page 173.

Q Define Credentials

Description	Enables you to add, remove and edit a credentials set for protocols.
Important Information	You configure a credentials set depending on what must be discovered and which protocols are supported on your site's network.
	For a list of protocols, see "Domain Credential References" on page 177.
	General information about the wizard is available in "Infrastructure Wizard" on page 128.
Wizard Map	The Infrastructure Discovery wizard contains: Infrastructure Wizard > Define IP Ranges > Define Credentials > Preferences > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
+	Add new connection details for selected protocol type.
×	Remove a protocol.
	Edit a protocol. Click to open the Protocol Parameters dialog box.
↑ ↓	Click a button to move a protocol up or down to set the order in which credential sets are attempted. DDM executes all the protocols in the list with the first protocol taking priority.
Protocol	Click to view details on the protocol, including user credentials.

2 Preferences

Description	Enables you to choose the configuration options to be used during discovery that are activated by the Infrastructure Discovery wizard.
Important Information	General information about the wizard is available in "Infrastructure Wizard" on page 128.
Wizard Map	The Infrastructure Discovery wizard contains: Infrastructure Wizard > Define IP Ranges > Define Credentials > Preferences > Schedule Discovery > Summary.

GUI Element (A-Z)	Description
DNS Nameservers	Discovers DNS nameserver machines and the IPs they hold names for.
	Choose to activate only if zone transfer can be performed from the Probe machine to the nameserver machines, that is, if the appropriate permissions exist on the DNS nameserver machines.
	Network implications. DDM tries to connect to DNS nameserver servers.
Failover Cluster Discovery	Discovers failover clusters including HP Service Guard, Microsoft Cluster Service and Veritas Cluster.

Chapter 4 • Run Discovery

GUI Element (A-Z)	Description
Host Information	Select the host resources to be discovered. These resources can be either physically or logically part of a host.
	After DDM connects to a host, it discovers the following resources:
	➤ For SNMP agents, the relevant Management Information Base (MIBs).
	➤ For WMI agents, the relevant Windows Management Instrumentation Query Language (WQL) queries.
	DDM can also execute shell commands on a machine.
	Network implications. The Software and Services network resources, because of the large quantities of data that they transmit, may cause very high network traffic. For this reason, the default is not to discover them.
Host TCP Connections	Discover TCP communication channels to map dependency relationships between hosts.
	This discovery requires that at least one protocol has a defined set of credentials. For details, see the previous Define Credentials step.
	Network implications.
	DDM executes shell commands on a machine to find open ports.

GUI Element (A-Z)	Description
IP Ping Strategy	Choose the strategy for discovering IPs in your environment.
	This discovery requires that the SNMP protocol be configured in the previous Define Credentials step.
	➤ Send ping request to every address in defined IP range.
	Select this option when you know that most of the IP addresses will respond, the network range is small, and most of the IPs in the range are of interest to you (that is, they are part of your network).
	Send ping request only to discoverable IPs in a network.
	Select this option when you know that not all IP addresses will respond and the network range is large. In this case, DDM first discovers a network, then sends a ping request to all discovered IPs in that network.
	Versions and Limitations.
	Verify that you have the correct credentials set for all the machines between the Probe and one of the network's switches.
Network Topology	Activate to discover the connections, on a discovered switch (for example, a host), between a host and its physical port as well as between a host and its logical layout (VLANs, ELANs).
	This discovery requires that at least one protocol has a defined set of credentials. For details, see the previous Define Credentials step.

GUI Element (A-Z)	Description
Port Scanning	The TCP ports appearing in the Choose TCP ports for port scanning list are scanned to discover open server ports. The ports are scanned on every discovered host.
	You can add new ports to be scanned, and you can remove existing ports from the list.
	To choose a port that does not appear in the list: 1 Click the Add port button to open the Add New Port dialog box.
	2 Click the Add port button and enter the port name and number.
	3 Click OK .
	Network implications.
	Note that the scanning process may affect performance on the network. Furthermore, you may need to inform machine owners that DDM will be trying to connect to their machines.
Software Element	Select to discover software elements running on the discovered hosts. As part of software element discovery, the processes and ports that are related to the software element are also discovered. The Software Library dialog box opens. For details, see "Software Library Dialog Box" on page 248.
	Network implications.
	A search pattern that is too general causes a toll on performance. For example, do not enter a process name that consists of an asterisk (*) only, as such a filter would try and retrieve all the processes running on all machines.



Description	Enables you to define a schedule for a specific job.
Important Information	For details on scheduling DDM, see "Discovery Scheduler Dialog Box" on page 124.
	General information about the wizard is available in "Infrastructure Wizard" on page 128.
Wizard Map	The Infrastructure Discovery wizard contains: Infrastructure Wizard > Define IP Ranges > Define Credentials > Preferences > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
%	You define a time template in the Discovery Scheduler pane of the Properties tab. For details, see "Discovery Scheduler Pane" on page 146.
Allow Discovery to run at	Choose the time at which the schedule should run.
Repeat Every	Select how often the schedule should run.

Summary

Description	Enables you to review the definitions before running discovery.
Important Information	Click Run to begin DDM. General information about the wizard is available in "Infrastructure Wizard" on page 128.
Wizard Map	The Infrastructure Discovery wizard contains: Infrastructure Wizard > Define IP Ranges > Define Credentials > Preferences > Schedule Discovery > Summary

12EE Wizard

Description	Enables you to run discovery on J2EE applications.
	To access: Admin > Universal CMDB > Discovery > Run Discovery > Basic Mode. Select the J2EE wizard from the list in the left pane. Click Configure and Run.
Important Information	For more information, hold the pointer over a question mark icon.
Wizard Map	The J2EE Discovery wizard contains: J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

Q Define Credentials

Description	Enables you to configure connection data for each protocol.
Important Information	You configure protocols depending on what must be discovered and which protocols are supported on your site's network.
	For a list of protocols, see "Domain Credential References" on page 177.
	General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains:
	J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
÷	Add new connection details for selected protocol type.
×	Remove a protocol.
⊘	Edit a protocol. Click to open the Protocol Parameters dialog box.
↑ ↓	Move a protocol up or down. DDM executes all the protocols in the list with the first protocol taking priority.
Protocol	Click to view details on the protocol, including user credentials. Note: A missing credential is represented by the following icon: Domains and Probes Domains and

🙎 J2EE Port Scanning

Description	Enables you to choose the port number and port type through which to connect to the J2EE application.
Important Information	General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains: J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
•	Click to add a port to the port list. The Add New Port dialog box opens. Select the ports and click OK.
	To edit existing system ports, in the Add New Port dialog box, click Edit Known System Ports . The Edit Known System Ports dialog box opens. Select the port and click the Edit button. In the dialog box that opens, make changes to the entries and click OK .
	To add a port to the list, in the Edit Known System Ports dialog box, click the Add button. Enter details of the port name, number and type and click OK .
×	Select a port and click the button to remove the port from the list.

Q WebLogic

Description	Enables you to select the JAR files for specific WebLogic versions.
Important Information	DDM supports the following WebLogic versions: 6.x, 7.x, 8.x, 9.x, and 10.x.
	1 To discover WebLogic, obtain the following drivers:
	➤ weblogic.jar (versions 6.x, 7.x, and 8.x only)
	➤ wlcipher.jar (if WebLogic is running on SSL, for all versions)
	➤ license.bea (if WebLogic is running on SSL but only for versions 6.x, 7.x, and 8.x)
	client trust store JKS file (for example, DemoTrust.jks, but only if WebLogic is running on SSL)
	➤ wlclient.jar (versions 9.x and 10.x only)
	➤ wljmxclient.jar (versions 9.x and 10.x only)
	2 Place the driver under the correct version folder in the following location:
	C:\hp\DDM\DiscoveryProbe\root\lib\collectors\pro beManager\discoveryResources\j2ee\weblogic\ <vers ion_folder>.</vers
	For example, C:\hp\DDM\DiscoveryProbe\root\lib\collectors\pro beManager\discoveryResources\j2ee\weblogic\6.x.
	3 Restart the Probe console before running the DDM jobs.
	4 On the WebLogic page in the J2EE wizard, select the check box for the versions to be discovered. Click Import file to open a browse window. Browse to the appropriate WebLogic JAR file, as listed below.
	General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains:
	J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

Chapter 4 • Run Discovery

GUI Element (A–Z)	Description
Activate using default JAR files (version 8.x only)	Select to enable discovery without specifying version- specific JAR files. This is less recommended and works on some environments only.
WebLogic version 6.x	 weblogic.jar For an SSL based discovery, select wlcipher.jar, license.bea, and the JKS file (for example, DemoTrust.jks)
WebLogic version 7.x	 weblogic.jar For an SSL based discovery, select wlcipher.jar, license.bea, and the client trust store JKS file (for example, DemoTrust.jks)
WebLogic version 8.x	 weblogic.jar For an SSL based discovery, select wlcipher.jar, license.bea, and the client trust store JKS file (for example, DemoTrust.jks)
WebLogic version 9.x	 wlclient.jar wljmxclient.jar For an SSL based discovery, select wlcipher.jar and the client trust store JKS file (for example, DemoTrust.jks)
WebLogic version 10.x	 wlclient.jar wljmxclient.jar For an SSL based discovery, select wlcipher.jar and the client trust store JKS file (for example, DemoTrust.jks)

Q WebSphere

Description	Enables you to select the JAR files for specific WebSphere versions.
Important Information	DDM supports the following WebSphere versions: 5.x, 6.0, and 6.1.
	➤ To discover WebSphere, obtain the following certificates:
	➤ client key store JKS file (DummyClientKeyFile.jks if WebSphere is running on SSL and the file is required)
	➤ client trust JKS file (DummyClientTrustFile.jks if WebSphere is running on SSL)
	Out-of-the-box drivers are located on the Probe machine at the following location: C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeManager\discoveryResources\j2ee\websphere
	➤ Restart the Probe console before running the DDM jobs.
	General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains:
	J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

Chapter 4 • Run Discovery

GUI Element (A–Z)	Description
Activate using default JAR files (versions 5.x, 6.x only)	Select to enable discovery without specifying version- specific JAR files. This is less recommended and works on some environments only.
WebSphere	Select the check box for the versions to be discovered. Click Import file to open a browse window. Browse to the appropriate WebSphere JAR file, as follows:
	➤ admin.jar
	➤ com.ibm.mq.pcf.jar
	➤ ffdc.jar
	➤ iwsorb.jar
	➤ j2ee.jar
	➤ jflt.jar
	➤ jmxc.jar
	➤ jmxx.jar
	➤ log.jar
	➤ mail.jar
	➤ ras.jar
	➤ sas.jar
	➤ security.jar
	➤ soap.jar
	➤ utils.jar
	➤ wasjmx.jar
	➤ websphere_arm_util.jar
	➤ wlmclient.jar
	➤ wsexception.jar
	➤ wssec.jar



Description	Enables you to select the JAR files for specific JBoss versions.
Important Information	DDM supports the following JBoss versions: 3.x, 4.x. General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains: J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
Activate using default JAR files (version 3.x, 4.x only)	Select to enable discovery without specifying version- specific JAR files. This is less recommended and works on some environments only.
JBoss version 3.x and 4.x	Select the check box for the versions to be discovered. Click Import file to open a browse window. Browse to the jbossall-client.jar JBoss JAR file.



Quantity Oracle Application Server

Description	Enables you to discover Oracle application servers.
Important Information	General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains: J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A-Z)	Description
Discover Oracle Application Server (version 10g)	Select to run DDM for the Oracle Application Server, version 10g.

Schedule Discovery

Description	Enables you to define a schedule for a specific job.
Important Information	General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains: J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary.

GUI Element (A–Z)	Description
%	You define a time template in the Discovery Scheduler pane of the Properties tab. For details, see "Discovery Scheduler Pane" on page 146.
Allow Discovery to run at	Choose the time at which the schedule should run.
Repeat Every	Select how often the schedule should run.

Summary

Description	Enables you to review the definitions before running discovery.
Important Information	To make changes to the run, click the Back button. General information about the wizard is available in "J2EE Wizard" on page 136.
Wizard Map	The J2EE Discovery wizard contains: J2EE Wizard > Define Credentials > J2EE Port Scanning > WebLogic > WebSphere > JBoss > Oracle Application Server > Schedule Discovery > Summary

GUI Element (A–Z)	Description
Run	Click to run DDM.

Properties Tab

Description	Enables you to view and administer the properties of modules and jobs. To access: Click the Properties tab in Run Discovery.
Important Information	Depending which level you select in the Discovery Modules pane, different information is displayed in the Properties tab. If you select:
	 The Discovery Modules root, all active jobs are displayed with scheduling information. Click any of the columns to sort the list by that column. Right-click a job to edit its scheduling. For details, see "Discovery Scheduler Dialog Box" on page 124. A Discovery module, the Description and Module Jobs panes are displayed. To edit a description, make changes in the Description pane and click OK.
	See also "Module Jobs Pane" on page 148. ➤ A job, the Parameters, Trigger TQLs, Global Configuration Files, and Discovery Scheduler panes are displayed. For details, see "Parameters Pane" on page 149, "Trigger TQLs Pane" on page 150, "Global Configuration Files Pane" on page 237, and "Discovery Scheduler Pane" on page 146.

Discovery Scheduler Pane

Description	Enables you to view information about the schedule set up for this job.
	To access: Select a job in the Discovery Modules pane in the Run Discovery window.

GUI Element (A–Z)	Description
	Click to add times to the Enable Discovery to run at list. The Time Templates dialog box opens. To add a time template to the list, in the Time Templates dialog box, click the Add button to open the Edit Time Template dialog box. For details, see "Edit Time Template Dialog Box" on page 126.
Edit Scheduler	Click to open the Discovery Scheduler. For details, see "Discovery Scheduler Dialog Box" on page 124.
Invoke on New Trigger CIs Immediately	A check mark signifies that the job runs as soon as the Trigger CI reaches the Probe. If this column does not contain a check mark, the job runs according to the schedule defined in the Schedule Manager.
Time Template	Choose a template that includes the days and times when the job should run.

Global Configuration Files Pane

For details, see "Global Configuration Files Pane" on page 237.

Module Jobs Pane

Description	Enables you to view the active jobs for a specific module.
	To access: Select a module in the Discovery Modules pane in the Run Discovery window.

GUI Element (A–Z)	Description
+	Add Discovery Job to Module. Opens the Choose Discovery Jobs dialog box where you can select jobs from more than one zip file. (Use the SHIFT or CTRL key to select several jobs.)
×	Remove Selected Discovery Job from Module. Select the job and click the button. (No message is displayed. To restore the job, click the Cancel button.)
<column title=""></column>	Click a column title to change the order of the CITs from ascending to descending order, or vice versa.
<list jobs="" of=""></list>	All jobs included in the module. (Displayed when a specific module is selected in the Discovery Modules pane.)
<right-click menu=""></right-click>	Right-click a row to open the Discovery Scheduler for the selected job. For details, see "Discovery Scheduler Dialog Box" on page 124.
	Right-click a column title to customize the table. Choose from the following options:
	 Hide Column. Select to hide a specific column. Show All Columns. Displayed when a column is hidden. Customize. Select to display or hide columns and to change the order of the columns in the table. Opens the Columns dialog box. Auto-resize Column. Select to change a column width to fit the contents. For details, see "Select Columns Dialog Box" in Reference Information.

GUI Element (A–Z)	Description
Invoke Immediately	 A check mark signifies that the Discovery job runs as soon as the triggered CI reaches the Probe. In this case, the Invoke on new triggered CIs immediately check box is selected in the Properties tab. If this column does not contain a check mark, the job runs according to the schedule defined in the Schedule Manager.
Job Name	The name of the job and the package in which the job is included. (Displayed when a job is selected in the Discovery Modules pane.)
Schedule Information	The scheduling information of the job as defined in the Discovery Scheduler.
Trigger TQLs	The name of the TQL that activated the job.

Parameters Pane

Description	Enables you to override pattern behavior.	
	To view a description, hold the pointer over the parameter.	
	To access: Select a job in the Discovery Modules pane in the Run Discovery window.	
Important Information	You can override a default pattern parameter for a specific job, without affecting the default value.	

Chapter 4 • Run Discovery

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description		
Name	The name given to the pattern.		
Override	Select to override the parameter value in the pattern. When this check box is selected, you can override the default value. For example, to change the protocolType parameter, select the Override check box and change MicrosoftSQLServer to the new value. Click OK to save the change:		
	Override	Name	Value
	Override .		MicrosoftSQLServer
	For details on editing Pattern Parameters Par	=	
Value	The value defined in the pattern.		

Trigger TQLs Pane

Description	Enables you to define one or more TQL queries to be used as triggers to activate the selected job.	
	To access:	
	➤ Select a job in the Discovery Modules pane in the Run Discovery window.	
	➤ Create a job by right-clicking a module in the Discovery Modules pane, and choose Create New Job.	

GUI Element (A–Z)	Description
+	Add TQL. You can add one or more non-default TQL queries to be used as triggers to activate the selected job. Click to open the Choose Discovery TQL dialog box.
×	Remove TQL. Select the TQL and click the button.
	(No message is displayed. To restore the TQL, click the Cancel button.)
	Note: If a TQL query is removed for an active job, DDM no longer receives new CIs coming from that TQL query. Existing Trigger CIs that originally came from the TQL query are not removed.
•••	Click to add or remove Probes for a specific TQL. For details, see "Edit Probe Limitation for TQL Output Dialog Box" on page 126.
⊘	Click to open the Trigger TQL Editor. For details, see "Trigger TQL Editor Window" on page 154.
17	Click to open the Query Manager. For details, see Chapter 9, "Query Manager."
Probe Limit	The Probes being used for the discovery process. To add or remove Probes, click the button.
TQL Name	The name of the trigger TQL query that activates the job.

Related Cls Window

Description	Enables you to view, in map format, the CIs that are related to a selected CI.
	To access: In the Discovered CIs dialog box, right-click a CIT and select Get Related CIs .
Important Information	Related CIs are CIs that are the parent, child, or sibling of an existing CI.

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A-Z)	Description
<right-click menu=""></right-click>	For details, see "IT Universe Manager Context Menu" in <i>Model Management</i> .
<menu></menu>	For details, see "Toolbar Options" in Model Management.
<topology map=""></topology>	For details, see "Topology Map Overview" in <i>Model Management</i> .

Show Results for Triggered CI Page

Description	Enables you to view the results of running an ad-hoc request to the Probe. DDM acquires the results by running the job on a selected trigger CI. In the case of an error, a message is displayed.
	To access: Run Discovery, select a module or job, select the Details tab. In the Discovery Status pane, drill down to a CI, right-click it, and choose Show Results for Triggered CI.

GUI Element (A-Z)	Description
3	Select a CIT and click to display additional information in the Sources CIs dialog box. For details, see "Source CIs Dialog Box" on page 153.
Q	Click to open a topology map showing a result map for the Triggered CI. Right-click a CIT to view its properties.

🙎 Source Cls Dialog Box

The Source CIs dialog box includes the same components as the **Discovered CIs** dialog box. For details, see "Discovered CIs Dialog Box" on page 119.

Time Templates Dialog Box

Description	Enables you to define a daily or weekly schedule to run selected jobs.
	To access: Run Discovery > Properties tab > Discovery Scheduler pane > Time Template icon.

GUI Element (A–Z)	Description
+	Click to add a time template. Opens the Edit Time Template dialog box.
×	Select a time template and click to delete.
⊘	Select a time template and click to edit it. Opens the Edit Time Template dialog box.

1 Triggered Cls

Description	Enables you to view all CI instances found for a selected TQL node.
	To access:
	➤ Run Discovery > Dependency Map tab. Right-click a CIT and select Show Triggered CIs.
	➤ In the Discovery Status pane, click the Show additional data button.
Important Information	The Triggered CIs window includes the same information as the Element Instances window. For details, see "Element Instances Dialog Box" in <i>Model Management</i> .

Trigger TQL Editor Window

Description	Enables you to edit a TQL that has been defined to trigger jobs. To access: Run Discovery > Properties tab > Trigger TQLs pane > select a TQL and click the Open the TQL Editor button.
Important Information	A Trigger TQL associated with a job is a subset of the Input TQL, and defines which specific CIs should be the Trigger CIs for a job. That is, if an Input TQL queries for IPs running SNMP, a Trigger TQL queries for IPs running SNMP in the range 195.0.0.0-195.0.0.10.
Useful Links	 ➤ "Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs" on page 50 ➤ "Input TQL Editor Window" on page 220

GUI Element (A–Z)	Description
<panes></panes>	➤ CI Type Selector Pane
	➤ Editing Pane
	➤ Information Pane
TQL Name	The name of the trigger TQL query that activates the job.

CI Type Selector Pane

Description	Displays a hierarchical tree structure of the CI Types found in the CMDB. For more details, see "CI Type Manager User Interface" in <i>Model Management</i> .
	Note : The number of instances of each CIT in the CMDB is displayed to the right of each CIT.
	To create or modify a TQL query, click and drag nodes to the Editing pane and define the relationship between them. Your changes are saved to the CMDB. For details, see "Add Nodes and Relationships to a TQL Query" in <i>Model Management</i> .
Included in Tasks	➤ "Define a TQL Query" in Model Management➤ "Create a Pattern View" in Model Management

Editing Pane

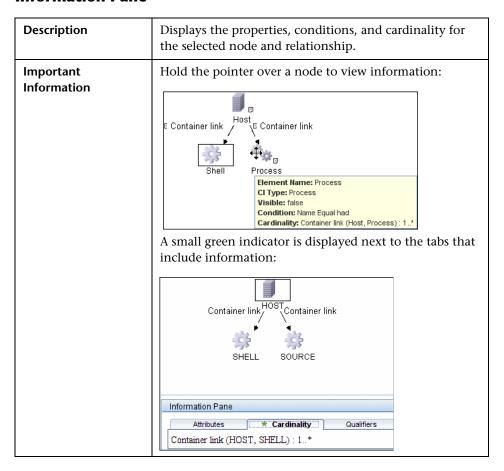
Description	Enables you to edit the node selected in the Trigger TQLs
	pane.

Chapter 4 • Run Discovery

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
<node></node>	Click to display information about the node in the information pane.
<right-click menu=""></right-click>	For details, see "Context Menu Options" in <i>Model Management</i> .
<toolbar></toolbar>	For details, see "Toolbar Options" in Model Management

Information Pane



GUI Element (A-Z)	Description
Edit	To view information, select a node or relationship in the Editing pane, select the tab in the Information Pane, and click the Edit button. For details on the Node Condition dialog box, see "Node/Relationship Properties Dialog Box" in <i>Model Management</i> .
Attributes	Displays the attribute conditions defined for the node or the relationship. For details, see "Attribute Tab" in <i>Model Management</i> .
Cardinality	Cardinality defines how many nodes you expect to have at the other end of a relationship. For example, in a relationship between host and IP, if the cardinality is 1:3, the TQL retrieves only those hosts that are connected to between one and three IPs. For details, see "Cardinality Tab" in <i>Model Management</i> .
Details	 ➤ CI Type. The CIT of the selected node/relationship. ➤ Visible. A tick signifies that the selected node or relationship is visible in the topology map. When the node/relationship is not visible, a box is displayed to the right of the selected node/relationship in the Editing pane: ➤ Include subtypes. Display both the selected CI and its descendants in the topology map.
Qualifiers	Displays the qualifier conditions defined for the node or the relationship. For details, see "Qualifier Tab" in <i>Model Management</i> .

Chapter 4 • Run Discovery

GUI Element (A-Z)	Description
Selected Identities	Displays the element instances that are used to define what should be included in the TQL results. For details, see "Identity Tab" in <i>Model Management</i> .

Set Up Discovery Probes

This chapter provides information on setting up Discovery and Dependency Mapping (DDM) Probes.

This chapter includes:

Concepts

➤ Job Execution Policies on page 159

Tasks

➤ Add a Probe on page 162

Reference

- ➤ Set Up Discovery Probes User Interface on page 163
- ➤ Domain Credential References on page 177

Job Execution Policies

You can define periods of time when a Probe must not run. You can choose to disable specific jobs running on any Probe or all jobs running on a specific Probe. You can also exclude jobs from a job execution policy so that they continue running as usual.

For details on defining a job execution policy, see "Add/Edit Policy Dialog Box" on page 166.

Example of Policy Ordering

There are two policies, **Total TCP Blackout** and **Always**. **Total TCP Blackout** does not allow any TCP discovery jobs to run. The policies appear in the list as follows:



A job (Class C IPs by ICMP) starts running. It checks the policies in the policy list from top to bottom. It starts by checking **Total TCP Blackout**. The job does not appear in this policy, so it continues down the list and checks **Always**. The job does appear here (**Allow All** is selected in the Edit Policy dialog box) so the job runs:



The next job (Software Element CF by Shell) starts running. It checks the policies in the policy list from top to bottom. It starts by checking **Total TCP Blackout**. The job appears in this policy (**Disallowed Jobs** is selected in the Edit Policy dialog box), so the job does not run:



Important: If a job is not connected to any policy, it does not run. To run these jobs, set the last policy in the list to **Allow All**.

Running Jobs When a Job Execution Policy Is Running

If a policy begins to operate while a Probe is executing a job, the job pauses. When the policy finishes, the job continues to run from where it ceased. For example, say a job contains 10,000 Trigger CIs. The job finishes working on 7,000 of them and then the policy starts to operate. When the job continues (after the policy finishes), it works on the remaining 3,000 Trigger CIs—the job does not start running from the beginning.

Add a Probe

This task describes how to add a Probe to DDM.

This task includes the following steps:

- ➤ "Prerequisites" on page 162
- ➤ "Add a Domain to DDM" on page 162
- ➤ "Add a Probe to the New Domain" on page 162
- ➤ "Add Further Probes to the Domain Optional" on page 162

1 Prerequisites

Verify that the Probe is installed and make a note of its IP address.

2 Add a Domain to DDM

In this step, you create the domain for the new Probe.

- Access the Set Up Discovery Probes window: Admin > Universal CMDB > Discovery > Set Up Discovery Probes.
- **b** Select **Domains and Probes** and click the **Add Domain or Probe** button to open the Add New Domain dialog box. For details, see "Add New Domain Dialog Box" on page 167.

3 Add a Probe to the New Domain

In this step, you define the Probe and its range.

- **a** Double-click the new domain and select the **Probes** folder.
- **b** Click the **Add Domain or Probe** button to open the Add New Probe dialog box. For details, see "Add New Probe Dialog Box" on page 168.
- **c** Select the new probe and define its IP range. For details, see "Add/Edit IP Range Dialog Box" on page 164.

4 Add Further Probes to the Domain – Optional

You can add more probes to this domain. For details, see the previous steps.

🔍 Set Up Discovery Probes User Interface

This section describes:

- ➤ Add/Edit IP Range Dialog Box on page 164
- ➤ Add/Edit Policy Dialog Box on page 166
- ➤ Add New Domain Dialog Box on page 167
- ➤ Add New Probe Dialog Box on page 168
- ➤ Choose Discovery Jobs Dialog Box on page 169
- ➤ Details Pane on page 169
- ➤ Domains and Probes Pane on page 173
- ➤ Edit Related Probes Dialog Box on page 174
- ➤ Edit Timetable Dialog Box on page 174
- ➤ Protocol Parameters Dialog Box on page 175
- ➤ Scope Definition Dialog Box on page 176
- ➤ Set Up Discovery Probes Window on page 176
- ➤ Selecting Probes on page 177

Name Add/Edit IP Range Dialog Box

Description	Enables you to set the network range for discovery. The results are retrieved from the addresses in the range you define. You can also define IP addresses that must be excluded from a range. To access: Click the Add IP range button in the Ranges pane (Admin > Universal CMDB > Discovery > Set Up Discovery Probes > Details pane).
Important Information	If you define a range that is out of the scope of the network on which the Probe is installed, HP Business Availability Center automatically defines the range of the IP on which the Probe is running. A message informs you that the Probe does not include the Probe range. Answer Yes to include the IP address in the range.
Included in Tasks	"Run Discovery – Advanced Mode Workflow" on page 74

GUI Element (A–Z)	Description
+	To exclude an IP range from discovery, click the Add IP range button.
×	To delete the excluded part of an IP range, select the excluded range and click the Remove IP range button.
	To edit the excluded part of an IP range, click the Edit IP range button. For details, see Exclude Ranges.

GUI Element (A–Z)	Description
Exclude Ranges	Click the Add IP range or Edit IP range button to exclude part of a range. In the Exclude IP Range dialog box, enter the range to exclude.
	Note:
	 Note: ➤ You must enter a range (in the Add/Edit IP Range dialog box) before you can enter the excluded range. ➤ The rules for entering an excluded range are the same as for entering a range. For details, see Range. ➤ Use this feature to divide a network range into several subranges. For example, say a range is 10.0.64.0 – 10.0.64.255. You define three excluded ranges: 10.0.64.45 – 10.0.64.50 10.0.64.65 – 10.0.64.70 10.0.64.89 – 10.0.64.95 Therefore, the ranges to be discovered are:
	10.0.64.0 – 10.0.64.44 10.0.64.51 – 10.0.64.64 10.0.64.71 – 10.0.64.88
	10.0.64.96 – 10.0.64.255

GUI Element (A–Z)	Description
Range	The rules for defining an IP address range are as follows:
	➤ The IP address range must have the following format: start_ip_address — end_ip_address
	For example: 10.0.64.0 - 10.0.64.57
	➤ The range can include an asterisk (*), representing any number in the range of 0-255.
	➤ If you use an asterisk, you do not need to enter a second IP address. For example, you can enter the range pattern 10.0.48.* to cover the range from 10.0.48.0 to 10.0.48.255.
	➤ Use an asterisk in the lower bound IP address of the IP range pattern only. (If you use an asterisk in the lower bound IP address and also enter an upper bound IP address, the upper bound IP address is ignored.)
	➤ You can use more than one asterisk (*) in an IP address as long as they are used consecutively. The asterisks cannot be situated between two numbers in the IP address, nor can they be substituted for the first digit in the number. For example, you can enter 10.0.*.* but not 10.*.64.*.
	➤ Two Probes in the same domain cannot have the same IP address in their range.

Add/Edit Policy Dialog Box

Description	Enables you to add a job execution policy, to disable jobs from running at specific times.
	To access: Admin > Universal CMDB > Discovery > Set Up Discovery Probes > Details pane > Job Execution Policy section. Select an existing policy and click Edit, or click the Add button.
Useful Links	"Job Execution Policies" on page 159 "Job Execution Policy Pane" on page 171 "Domain Credential References" on page 177

GUI Element (A–Z)	Description
Related jobs	 Allow all. Run the job execution policy on all jobs. Total blackout. The policy does not run on any jobs. Allowed jobs. Choose jobs to run even during the configured blackout time. Disallowed jobs. Choose jobs that do not run during the configured blackout time. For allowed and disallowed jobs, click the Add job or Remove job button to choose specific jobs to be included in, or excluded from, the policy. If you click the Add job button, the Choose Discovery Jobs dialog box opens.
Related Probes	The Probes on which to run the policy. Click the button to open the Edit Related Probes dialog box to define which Probes are included in the policy.
Time	The date and time during which the policy is active. Click the button to open the Edit Timetable dialog box.

Add New Domain Dialog Box

Description	Enables you to add a domain.
	To access: Click the Add Domain or Probe button in the Domains and Probes pane.
Important Information	In a version 8.00 environment that has been upgraded from version 6.x, to enable data to be modelled similarly as in the previous version, you must define the Probes as belonging to the External domain and not to the Customer domain.

Chapter 5 • Set Up Discovery Probes

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Description	Enter a description to appear in the Details pane of the Set Up Discovery Probes window.
Domain Type	 Customer. A private domain used for your site. You can define several domains and each domain can include multiple Probes. Each Probe can include IP ranges but the customer domain itself has no range definition. External. Internet/public domain. A domain that is defined with a range. The external domain can contain only one Probe whose name equals the domain name. However, you can define several external domains in your system.
Name	Enter a unique name for the domain.

New Probe Dialog Box

Description	Enables you to add a Probe. To access: Click the Add Domain or Probe button in the Domains and Probes pane.
Important Information	 To add a Probe to an existing domain, select Probes in the Domains and Probes pane and click the Add Domain or Probe button. To add a Probe to a new domain, create a domain, then add the Probe to the domain. Two Probes in the same domain cannot have the same IP address in their range. When a Probe is activated, it is added automatically and its status changes to connected. For details, see "Launch the Probe" on page 37.

Choose Discovery Jobs Dialog Box

Description	Enables you to choose the jobs that are to be added to, or excluded from, the job execution policy.
	To access: Select Allowed Jobs or Disallowed jobs in the Edit Policy dialog box and click the button .

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
<installed packages=""></installed>	Locate the job to be included in, or excluded from, the policy. To select jobs from several packages, hold down the CTRL key and make the selection.

Q Details Pane

Description	Enables you to view the Probes running under all domains and to add an execution policy to jobs (that is, to schedule time periods when jobs should not run). To access: Click an object in the Domains and Probes pane.
Important Information	Depending on what you select in the Domains and Probes pane, different information is displayed in the Details tab. For details, see "Displayed Information" on page 170 in the next section.

Displayed Information

If You Select	The Information Displayed Is
Domains and Probes	Domains and Probes, you can view details on all Probes and you can define and edit job execution policies. For details, see "Discovery Probes Pane" on page 170 and "Job Execution Policy Pane" on page 171.
Domains and Probes Domains and Probes Domains and Probes DefaultDomain Credentials	A specific domain, you can add a description and view a list of Probes running in that domain. For details, see "Discovery Probes Pane" on page 170 and "Description Pane" on page 172.
Domains and Probes Domains and Probes Domains and Probes Domains and Probes Credentials DOMAINS Protocol	A specific protocol, you can add protocol parameters and you can view details on the protocol, including user credentials. For details, see "Domain Credential References" on page 177.
Probes ((*) LABMAM1	A specific Probe, you can view details on the Probe, including range information. You can also add ranges to, or exclude ranges from, the Probe. For details, see "Ranges Pane" on page 172, "Discovery Probes Pane" on page 170, and "Details Pane" on page 172.

Discovery Probes Pane

The Discovery Probes pane includes a list of all Probes connected to the server. The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
IP	The IP range defined during Probe creation.
Last Access Time	The last time that the Probe requested tasks from the server.
Name	The Probe name as it appears in DDM.
Status	Can be Connected or Disconnected.

Job Execution Policy Pane

Description	Enables you to configure the periods of time when jobs should not run.
	To access: Admin > Universal CMDB > Discovery > Set Up Discovery Probes. Select Domains and Probes.
Important Information	Jobs that have a listening functionality (that is, they do not perform discovery, for example, they listen to SNMP traps) are not included in a policy.
Useful Links	"Job Execution Policies" on page 159 "Domain Credential References" on page 177

GUI Element (A–Z)	Description
₩ ₩	Move the policy up or down. DDM executes all the policies in the list with the first policy taking priority. If a job is included in two policies, DDM executes the first policy only for that job.
+	Add a policy.
×	Remove a policy.
⊘	Edit a policy. Click to open the Edit Policy dialog box.
Jobs	The jobs that are affected by the policy.
Probes	The Probes that are affected by the policy.
Time	The schedule of the policy.

Description Pane

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Description	The description that was entered during domain creation.
Domain Type	For details, see Domain Type in "Add New Domain Dialog Box" on page 167.

Details Pane

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Last time Probe accessed	The last time that the Probe was accessed on the server machine.
Probe IPs	The IP of the Probe machine.
Status	 Connected. The Probe has successfully connected to the server (the Probe connects every few seconds). Disconnected. The Probe is not connected to the server.

Ranges Pane

GUI Element (A–Z)	Description
+	Click to open the Add IP Range dialog box.
×	Click a range and click the button to remove a range from the list.
⊘	Click to open the Edit IP Range dialog box.

GUI Element (A–Z)	Description
	Export a permission object in Excel, PDF, RFT, CSV, or XML format. For details, see "Browse Mode" in <i>Model Management</i> .
	Click to import ranges from a CSV file. Before using this feature, verify that the imported file is a valid CSV file, and that the ranges in the file do not conflict with existing ranges (that is, there are no duplicate or overriding ranges).
Excluded	Displays the IP addresses that have been excluded from the range that the Probe uses to discover CIs. For details, see "Add/Edit IP Range Dialog Box" on page 164.
Range	The network IP addresses that the Probe uses to discover CIs. For details, see "Add/Edit IP Range Dialog Box" on page 164.

Domains and Probes Pane

Description	Enables you to view, define, or edit a domain, a Probe or a Probe's credentials.
	To access: Admin > Universal CMDB > Discovery > Set Up Discovery Probes.
Important Information	A missing credential is represented by the following icon: Domains and Probes Domains and Probes Domains and Probes Credentials DefaultDomain DefaultDomain DefaultDomain DefaultDomain DefaultDomain SAP Protocol SAP JMX Protocol SAP JMX Protocol SAP Frotocol SAP Trotocol SAP Trotocol Massing credentials Dud Missing credentials
Useful Links	"Job Execution Policies" on page 159

Chapter 5 • Set Up Discovery Probes

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
*	Adds a domain or Probe, depending on what is selected.
×	Deletes a domain or Probe, depending on what is selected.
0	Updates all domain and Probe information.

🔍 Edit Related Probes Dialog Box

Description	Enables you to select specific Probes.
	To access: Click the Related Probes button in the Edit Policy dialog box.
Useful Links	"Job Execution Policies" on page 159

Lesson Edit Timetable Dialog Box

Description	Enables you to set the times when a Probe must run a job execution policy. To access: Click the Edit button in the Edit Policy dialog box.
Useful Links	"Add/Edit Policy Dialog Box" on page 166

GUI Element (A–Z)	Description			
Description	Add a description of the specific policy. This field is mandatory.			
	Tip: The text you enter here appears in the Time box in the Job Execution Policy pane, so it is recommended that the description be informative:			
	Job Execution Policy ↑ ↓ ↑ × ℓ			
	Time	Probes	Jobs	
	Labor Day weekend	All	None (total blackout)	
	Always	All	All	
Time Definition	Click a cell for a day and time to be included in the policy. To add more than one time unit, drag the pointer over the cells.			
	Note: To clear a ti	me unit, click t	he cell a second tim	e.

Protocol Parameters Dialog Box

Description	Displays the attributes that can be defined for a protocol.
	To access: Set Up Discovery Probes > Domains and Probes > Domain > Credentials, select a protocol and click the Add or Edit button.
Important Information	For the description of each protocol, see "Domain Credential References" on page 177.

Scope Definition Dialog Box

Description	Enables you to set the range that a protocol must discover.
	To access: Click the Edit button in the Protocol Parameters dialog box.

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Selected Probes	To select specific Probes whose IP range must be changed, click Edit . For details, see "Choose Probe Dialog Box" on page 98.
Selected Ranges	➤ All. The protocol runs discovery on all ranges for the domain.
	➤ Selected Range. For the procedure to select a specific range on which the protocol runs discovery or to define an excluded range, see "Add/Edit IP Range Dialog Box" on page 164.

Set Up Discovery Probes Window

Description	Enables you to define a new domain or to define a new Probe for an existing domain. Also, to define the connection data for each protocol.
	To access: Admin > Universal CMDB > Discovery > Set Up Discovery Probes.
Important Information	 ➤ For details on the Domains and Probes pane, see "Domains and Probes Pane" on page 173. ➤ For details on the Details pane, see "Details Pane" on page 169.
Useful Links	"Domain Credential References" on page 177

Selecting Probes

The Choose Probe to Filter, Edit Probe Limitations for TQL Output, and Edit Related Probes dialog boxes include the following elements (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
	Add Selected Probe. Click to add a Probe to the Selected Probes column.
<	Remove Selected Probe. Click to remove a Probe from the Selected Probes column.
All Discovery Probes	 ➤ Select to add all Probes in the Non-selected Probes list. ➤ Clear to add a specific Probe from the Non-selected Probes list.
Non-selected Probes	Probes that are not included in the policy/filter/limitations.
Selected Probes	Probes that are included in the policy/filter/limitations.

Q Domain Credential References

This section explains protocol credentials. You can edit credential attributes. For details, see "Protocol Parameters Dialog Box" on page 175.

When a protocol is selected in the Domains and Probes Pane, the following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A-Z)	Description
+	Click to add new connection details, to open the Add Protocol Parameters dialog box.
×	Select a protocol and click to remove connection details. Answer OK to the message.
⊘	Select a protocol and click to edit connection details in the Protocol Parameters dialog box.

GUI Element (A-Z)	Description
↑	Select a protocol and click an arrow to move the protocol instance up or down.
	The order of the policies in the list defines which policy is checked first: a job starts running and checks the policy list from top to bottom. If the job name exists in a policy, the job runs. For details on adding jobs to a protocol, see "Add/Edit Policy Dialog Box" on page 166. For details on job execution policies, see "Example of Policy Ordering" on page 160.
<right-click a<="" th=""><th>Choose from the following options:</th></right-click>	Choose from the following options:
credential>	➤ Edit. Choose this option to enter protocol parameters, such as user name and password, that enable DDM to connect to an application on a remote machine.
	➤ Edit using previous interface. Choose this option if:
	➤ In a previous version of HP Business Availability Center, you added parameters to this protocol that do not exist in this version.
	➤ Values in this version cannot be deleted. For example, in this version you cannot configure SQL Protocol credentials with an empty port number. Select this option to open the previous Edit Protocol Parameter dialog box and delete the port number.
	➤ Check credentials. In the box that opens, enter the IP address of the remote machine on which the protocol must run. The Probe attempts to connect to this IP and returns an answer as to whether the connection succeeded or not.
<right-click a="" title=""></right-click>	Choose from the following options:
	➤ Hide Column. Displayed when a column is shown.
	➤ Show All Columns. Displayed when a column is hidden.
	➤ Customize. Select to change the display order of the columns.
	➤ Auto-resize Column. Select to change the column width to fit the contents.

All protocol credentials include the following parameters:

Parameter	Description
Index	Indicates the order in which protocol instances are used to make a connection attempt. The lower the index, the higher the priority.
	Default: 9999 . If you do not change the default, this protocol instance is used last.
Network Scope	To change the range that a protocol must discover or to select a Probe, click Edit . For details, see "Scope Definition Dialog Box" on page 176. Default: ALL .
User Label	Enter a label to help you identify a specific protocol credential, when you use it later. Enter a maximum of 50 characters.

This section includes the following topics:

- ➤ "JBoss Protocol" on page 180
- ➤ "NNM Protocol" on page 180
- ➤ "NTCMD Protocol" on page 181
- ➤ "SAP JMX Protocol" on page 181
- ➤ "SAP Protocol" on page 182
- ➤ "Siebel Gateway Protocol" on page 183
- ➤ "SNMP Protocol" on page 183
- ➤ "SQL Protocol" on page 185
- ➤ "SSH Protocol" on page 186
- ➤ "Telnet Protocol" on page 188
- ➤ "UDDI Registry Protocol" on page 190
- ➤ "VMware Infrastructure Management Protocol (VIM)" on page 190
- ➤ "WebLogic Protocol" on page 191
- ➤ "WebSphere Protocol" on page 193
- ➤ "WMI Protocol" on page 194



1 JBoss Protocol

Parameter	Description
Port Number	The port number.
Connection Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the JBoss application server.
User Name	The name of the user needed to connect to the application.
Password	The password of the user needed to connect to the application.



NNM Protocol

Parameter	Description
Connection Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the NNM server.
NNM Password	The password for the NNM Web service (for example, Openview).
NNM Server	The host name or IP address of the NNM server.
NNM User name	The user name of the NNM Web service (for example, system).
NNM Webservice Port	The Web service port number of the NNM server (for example, 80).
UMCBD Password	The password for the UCMDB Web service (the default is admin).
UCMDB Server	The host name or IP address of the UCMDB server.
UCMDB Username	The user name of the UCMDB Web service (the default is admin).
UCMDB Webservice Port	The UCMDB Web service port number (the default is 8080).
UCMDB Webservice Protocol	The protocol for the UCMDB Web service (the default is http).

NTCMD Protocol

Parameter	Description
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the NTCMD server.
User Name	The name of the user needed to connect to the host as administrator.
Password	The password of the user needed to connect to the host as administrator.
Windows Domain	The name of the domain that includes the host where the Probe is installed.

💐 SAP JMX Protocol

Parameter	Description
Port Number	The SAP JMX port number. The SAP JMX Port structure is usually 5 <system number="">04. For example, if the system number is 00, the port is 50004.</system>
	Leave this field empty to try to connect to the discovered SAP JMX port; SAP JMX port numbers are defined in the portNumberToPortName.xml configuration file.
Connection Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the SAP JMX console.
User Name	The name of the user needed to connect to the application as administrator.
Password	The password of the user needed to connect to the application as administrator.

SAP Protocol

Parameter	Description	
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the SAP console.	
User Name		eded to log in to the SAP system.
	The user should have the	e following permissions:
	Authorization Object	Authorization
	S_RFC	*
	S_XMI_PROD	EXTCOMPANY=MERCURY;EXTPROD UCT=DARM;INTERFACE=XAL
	S_TABU_DIS	DICBERCLS=SS; DICBERCLS=SC
Password	The password of the user needed to log in to the SAP system.	
SAP Client Number	It is recommended to use the default value (800).	
SAP System Number	It is recommended to us	e the default value (00).
SAP Router String	A route string describes the connection required between two hosts using one or more SAProuter programs. Each of these SAP router programs checks its Route Permission Table (http://help.sap.com/saphelp_nw04/helpdata/en/4f/992dfe446 d11d189700000e8322d00/content.htm) to see whether the connection between its predecessor and successor is allowed. If it is, SAProuter sets it up.	



Parameter	Description
Connection Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the Siebel Gateway console
User Name	The name of the user needed to log on to the Siebel enterprise
Password	The password of the user needed to log on to the Siebel enterprise.
Siebel Site Name	The name of the Siebel Enterprise.
Path to Siebel Client	The location on the Probe server where you copied srvrmgr. For details, see "Prerequisites – Copy the driver Tool to the Probe Server" on page 284.
	Note: If there are several protocol entries with different srvrmgr versions, the entry with the newer version should appear before the entry with the older version. For example, to discover Siebel 7.5.3. and Siebel 7.7, define the protocol parameters for Siebel 7.5 and then the protocol parameters for Siebel 7.5.3.

SNMP Protocol

Parameter	Description
Port	(For SNMP versions v1, v2, and v3) The port number on which the SNMP agent listens.
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the SNMP agent.
Retry	The number of times the Probe tries to connect to the SNMP agent. If the number is exceeded, the Probe stops attempting to make the connection.

Parameter	Description
Versions 1, 2	Community. Enter the authentication password you used when connecting to the SNMP service community (which you defined when configuring the SNMP service—for example, a community for read-only or read/write).
Version 3	Authentication method : Select one of the following options for securing the access to management information:
	➤ NoAuthNoPriv. Using this option provides no security, confidentiality, or privacy at all. It can be useful for certain applications, such as development and debugging to turn security off. This option requires only a user name for authentication (similar to requirements for v1 and v2).
	➤ AuthNoPriv. The user logging on to the management application is authenticated by the SNMP v3 entity before the entity allows the user to access any of the values in the MIB objects on the agent. Using this option requires a user name, password, and the authentication algorithm (HMAC-MD5 or HMAC-SHA algorithms).
	➤ AuthPriv. The user logging on to the management application is authenticated by the SNMP v3 entity before the entity allows the user to access any of the values in the MIB objects on the agent. In addition, all of the requests and responses from the management application to the SNMP v3 entity are encrypted, so that all the data is completely secure. This option requires a user name, password, and an authentication algorithm (either (HMAC-MD5 or HMAC-SHA).
	User Name : The name of the user authorized to log on to the management application.
	Password : The password used to log on to the management application.
	Authentication algorithm : The MD5 and SHA algorithms are supported.
	Privacy key : The secret key used to encrypt the scoped PDU portion in an SNMP v3 message.
	Privacy algorithm : The DES algorithm is supported.

💐 SQL Protocol

Parameter	Description
Database Type	The database type. Select the appropriate type from the box.
Port	The port number on which the database server listens.
	➤ If you enter a port number, DDM tries to connect to a SQL database using this port number.
	➤ For an Oracle database: If there are many Oracle databases in the environment and you do not want to have to create a new credential for each separate database port, you leave the Port Number field empty. When accessing an Oracle database, DDM refers to the portNumberToPortName.xml file and retrieves the correct port number for each specific Oracle database port.
	Note : You can leave the port number empty on condition that:
	 ➤ All Oracle database instances are added to the portNumberToPortName.xml file. For details, see "portNumberToPortName.xml" on page 204. ➤ The same user name and password is needed to access all Oracle database instances.
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the database.
User Name	The name of the user needed to connect to the database.
Password	The password of the user needed to connect to the database.
Database Name	The database name.

SSH Protocol

Parameter	Description
Port	By default an SSH agent uses port 22. If you are using a different port for SSH, enter that port number.
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the remote machine.
	For the UNIX platform: If your server is slow, it is recommended to change Timeout to 40000.
Version	SSH2. Connect through SSH-2 only.
	SSH1. Connect through SSH-1 only.
	SSH2 or SSH1. Connect through SSH-2 and in case of error (if SSH-2 is not supported by the server), try to connect through SSH-1.
Shell Command Separator	The character that separates different commands in a shell (to enable the execution of several commands in the same line). For example, in UNIX, the default shell command separator is a semicolon (;). In Windows, the shell command separator is a double ampersand (&&).
Authentication Method	Choose one of the following authentication options to access SSH:
	 password. Enter a user name and password. publickey. Enter the user name and path to the key file that authenticates the client. keyboard-interactive. Enter questions and answers. For details, see "Prompts and Responses" on page 187.
User Name	The name of the user needed to connect to the host through the SSH network protocol.
Password	The password of the user needed to connect to the host.
Key File Path	(Enabled when the publickey authentication method is selected.) Location of the authentication key. (In certain environments, the key path is required to connect to an SSH agent.)
	Note : Enter the complete path to the key file on the Probe machine.

or more prompts to enter information and the client displays them and sends back responses keyed-in by the composed them are seponses and expected responses: Prompt: Please enter your user name. Response: Shelly-Ann Prompt: What is your age? Response: 21 Prompt: This computer is HP property. Press y to enter. Response: y To create these prompts and responses, enter the following strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y	Parameter	Description
responses: Prompt: Please enter your user name. Response: Shelly-Ann Prompt: What is your age? Response: 21 Prompt: This computer is HP property. Press y to enter. Response: y To create these prompts and responses, enter the followistrings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH pronfor example: Authentication Method keyboard-interactive User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths	-	method is selected.) A method whereby the server sends one
Response: Shelly-Ann Prompt: What is your age? Response: 21 Prompt: This computer is HP property. Press y to enter. Response: y To create these prompts and responses, enter the followistrings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH promfor example: Authentication Method keyboard-interactive User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		
Prompt: What is your age? Response: 21 Prompt: This computer is HP property. Press y to enter. Response: y To create these prompts and responses, enter the following strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH prometor example: Authentication Method User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		Prompt : Please enter your user name.
Response: 21 Prompt: This computer is HP property. Press y to enter. Response: y To create these prompts and responses, enter the following strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH prone for example: Authentication Method User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		Response: Shelly-Ann
Prompt: This computer is HP property. Press y to enter. Response: y To create these prompts and responses, enter the following strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH promotion for example: Authentication Method User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		Prompt: What is your age?
Response: y To create these prompts and responses, enter the following strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH prometor example: Authentication Method keyboard-interactive User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		Response: 21
To create these prompts and responses, enter the following strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH promotor example: Authentication Method User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		Prompt : This computer is HP property. Press y to enter.
strings in the fields, separated by commas: Prompts: user,age,enter Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH pronfor example: Authentication Method		Response: y
Response: Shelly-Ann,21,y You can enter the full string as it appears in the SSH pron for example: Authentication Method Keyboard-Interactive Value Va		To create these prompts and responses, enter the following strings in the fields, separated by commas:
You can enter the full string as it appears in the SSH pron for example: Authentication Method Keyboard-Interactive Value Val		Prompts: user,age,enter
for example: Authentication Method keyboard-interactive V User Name Password		Response: Shelly-Ann,21,y
User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths		You can enter the full string as it appears in the SSH prompt, for example:
OK Cancel Help		User Name Password Key File Path Prompts Please enter your user name Responses Sudo paths Sudo commands

Chapter 5 • Set Up Discovery Probes

Parameter	Description
Sudo paths	The full paths to the sudo command. Paths are separated by commas.
Sudo commands	A list of commands that can be executed with the sudo command. Commands are separated by commas. For all commands to be executed with sudo, add an asterisk (*) to this field.

Telnet Protocol

Parameter	Description
Port	The port number. By default a Telnet agent uses port 23. If you are using a different port for Telnet in your environment, enter the required port number.
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the remote machine. For UNIX platforms: If your server is slow, it is recommended to change Connection Timeout to 40000.
Authentication Method	 Choose one of the following authentication options to access Telnet: password. Enter a user name and password. publickey. Enter the user name and path to the key file that authenticates the client. keyboard-interactive. Enter questions and answers. For details, see "Prompts and Responses" on page 187.
User Name	The name of the user needed to connect to the host.
Password	The password of the user needed to connect to the host.

Parameter	Description
Prompts and Responses	(Enabled when the keyboard-interactive authentication method is selected.) A method whereby the server sends one or more prompts to enter information and the client displays them and sends back responses keyed-in by the user.
	The following is an example of prompts and expected responses:
	Prompt : Please enter your user name.
	Response: Shelly-Ann
	Prompt : What is your age?
	Response: 21
	Prompt : This computer is HP property. Press y to enter.
	Response: y
	To create these prompts and responses, enter the following strings in the fields, separated by commas:
	Prompts: user,age,enter
	Response: Shelly-Ann,21,y
	You can enter the full string as it appears in the Telnet prompt, for example:
	Authentication Method keyboard-interactive User Name Password Key File Path
	Prompts Please enter your user name
	Responses
	Sudo paths
	Sudo commands
	or you can enter a key word, for example, user . DDM maps this word to the correct prompt.

Parameter	Description
Sudo paths	The full paths to the sudo command. Paths are separated by commas.
Sudo commands	A list of commands that can be executed with the sudo command. Commands are separated by commas. For all commands to be executed with sudo, add an asterisk (*) to this field.

UDDI Registry Protocol

Parameter	Description
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the UDDI Registry.
UDDI Registry URL	The URL where the UDDI Registry is located.

VMware Infrastructure Management Protocol (VIM)

Parameter	Description
Connection Timeout	Time-out in milliseconds after which the Probe stops trying to connect to VMware Infrastructure.
Port Number	DDM uses the number defined here when processing one of the Network – VMware jobs:
	If the port number is left empty, DDM performs a WMI query to extract the port number from the registry. DDM queries HKLM\SOFTWARE\VMware, Inc.\VMware VirtualCenter and searches for the HttpsProxyPort or HttpProxyPort attributes:
	➤ If the HttpsProxyPort attribute is found, DDM uses its value for the port and sets the prefix to HTTPS.
	➤ If the HttpProxyPort attribute is found, DDM uses its value for the port and sets the prefix to HTTP.
Use SSL	true : DDM uses a Secure Sockets Layer (SSL) protocol to access VMware Infrastructure, and the prefix is set to HTTPS .
	false: DDM uses the http protocol.

Parameter	Description
User Name	The name of the user needed to connect to VMware Infrastructure.
User Password	The password of the user needed to connect to VMware Infrastructure.

WebLogic Protocol

Parameter	Description
Port Number	 ➤ If you enter a port number, DDM tries to connect to WebLogic using this port number. ➤ However, say you know that there are many WebLogic machines in the environment and do not want to have to create a new credential for each machine. You leave the Port Number field empty. When accessing a WebLogic machine, DDM refers to the WebLogic port (defined in portNumberToPortName.xml) already found on this machine (by TCP scanning, using the Network Connection – Active Discovery module). Note: You can leave the port number empty on condition that: ➤ All WebLogic ports are added to the portNumberToPortName.xml file. For details, see "portNumberToPortName.xml" on page 204. ➤ The same user name and password is needed to access all WebLogic instances.
Connection Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the WebLogic application server.
User Name	The name of the user needed to connect to the application.
Password	The password of the user needed to connect to the application.
Protocol	An application-level protocol that determines whether DDM should connect to the server securely. Enter either http or https.

Chapter 5 • Set Up Discovery Probes

Parameter	Description
Trust Store File	The name of the SSL trust store file.
Name	To use the trust store file, do one of the following:
	➤ Enter the name (including the extension) and place the file in the following Discovery resources folder: <pre> <discovery directory="" installation="" probe="">\root\lib\collectors\probeManager\discovery Resources\j2ee\weblogic\<weblogic version="">.</weblogic></discovery></pre>
	➤ Insert the trust store file full path.
Trust Store Password	The SSL trust store password.
Key Store File	The name of the SSL keystore file.
Name	To use the keystore file, do one of the following:
	 Enter the name (including the extension) and place the file in the following Discovery resources folder: <discovery installation<="" p="" probe=""> directory>\root\lib\collectors\probeManager\discovery Resources\j2ee\weblogic\<weblogic version="">. </weblogic></discovery> Insert the keystore file full path.
Key Store Password	The password for the keystore file.

WebSphere Protocol

Parameter	Description
Port	The protocol port number as provided by the WebSphere system administrator.
	You can also retrieve the protocol port number by connecting to the Administrative Console using the user name and password provided by the WebSphere system administrator.
	In your browser, enter the following URL: http:/ <host>:9060/admin, where:</host>
	➤ <host> is the IP address of the host running the WebSphere protocol</host>
	➤ 9060 is the port used to connect to the WebSphere console
	Access Servers > Application Servers > Ports > SOAP_CONNECTOR_ADDRESS to retrieve the required port number.
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the WebSphere server.
User Name	The name of the user needed to connect to the application.
Password	The password of the user needed to connect to the application.
Trust Store File	The name of the SSL trust store file.
Name	To use the trust store file, do one of the following:
	 Enter the name (including the extension) and place the file in the following Discovery resources folder: <discovery installation<="" p="" probe=""> directory>\root\lib\collectors\probeManager\discover yResources\j2ee\websphere. </discovery> Insert the trust store file full path.
Trust Store Password	The SSL trust store password.

Chapter 5 • Set Up Discovery Probes

Parameter	Description
Key Store File Name	The name of the SSL keystore file. To use the keystore file, do one of the following: ➤ Enter the name (including the extension) and place the file in the following Discovery resources folder: <discovery directory="" installation="" probe="">\root\lib\collectors\probeManager\discover yResources\j2ee\websphere. ➤ Insert the keystore file full path.</discovery>
Key Store Password	The password for the keystore file.

WMI Protocol

Parameter	Description
Timeout	Time-out in milliseconds after which the Probe stops trying to connect to the WMI agent.
User Name	The name of the user needed to connect to the host.
Password	The password of the user needed to connect to the host.
Windows Domain	The name of the domain to which the user needed for the connection belongs.

Manage Discovery Resources

This chapter provides information on managing Discovery and Dependency Mapping resources such as patterns and configuration files.

This chapter includes:

Concepts

- ➤ Automatically Deleted System Components on page 196
- ➤ Creating a Discovery Wizard on page 198
- ➤ Discovering Software Elements on page 198

Tasks

- ➤ Configure the DDM Probe to Automatically Delete CIs Workflow on page 199
- ➤ Discover Software Elements Scenario on page 200

Reference

- ➤ Resource Files on page 204
- ➤ Manage Discovery Resources User Interface on page 208

Automatically Deleted System Components

During discovery, the DDM Probe compares CIs found during the previous, successful invocation with those found during the current invocation. A missing component, such as a disk or software, is assumed to have been removed from the system, and its CI is deleted from the Probe's database. The DDM Probe does not wait for the aging mechanism to perform the calculation but immediately sends a deletion request to the server.

You can define that CI instances are to be deleted for specific jobs. For details, see "Configure the DDM Probe to Automatically Delete CIs -Workflow" on page 199.

By default, the DDM Probe deletes the CI instances of the following CITs:

Job	Pattern	CIT/Link Type
Host Connection By SNMP	SNMP_NET_Dis_Connection	Contained, SNMP
Host Connection By Shell	Host_Connection_By_Shell	Contained, Shell
Host Connection By WMI	WMI_NET_Dis_Connection	Contained, WMI
Host Resources by SNMP	SNMP_HR_All	Disk, Service, Software, OS User
Host Resources by WMI	WMI_HR_All	Disk, Service, Software, OS User
Host Resources by Shell	TTY_HR_All	Disk, Service, Software, OS User, dir

Note:

- ➤ The change is defined on the job's pattern.
- ➤ If discovery fails and errors occur, no objects are sent for deletion.
- ➤ Carefully choose the CIs that are to be candidates for deletion. For example, process CITs are not good candidates because they are often shutting down and starting up again and as a result may be deleted at every invocation.
- ➤ You can use this procedure to delete relationships, too. For example, the contained relationship is used between a host and an IP address. A laptop machine is allocated a different IP address very often; by deleting the relationship, you prevent the accumulation of old IP addresses attached to this host.

Example of Automatic Deletion

During the previous invocation, the DDM Probe ran the Host Resources by WMI job and discovered a host with disks a, b, c, and d. During the current invocation, the Probe discovers disks a, b, and c, compares this result with the previous result, and deletes the CI for disk d.

More Information

- ➤ You can view deleted CIs in the Probe log and in the Deleted column in the Statistics Results pane. For details, see "Probe Logs" on page 57 and "Statistics Results Pane" on page 117.
- ➤ For details on aging, see the Results Management pane in "Pattern Management Tab" on page 228.
- ➤ An invocation is run according to the Scheduler. For details, see "Discovery Scheduler Dialog Box" on page 124.

Creating a Discovery Wizard

As the creation of Discovery wizards entails a very advanced knowledge of DDM, we recommend that you contact HP Software Support before beginning.

Discovering Software Elements

You can discover software (for example, a specific Oracle database) running in your environment.

This section includes the following topics:

- ➤ "Discovery Process" on page 198
- ➤ "Running Jobs" on page 199
- ➤ "Software Elements Default Views" on page 199

Discovery Process

The discovery process runs as follows:

- ➤ The Software Element jobs are activated.
- ➤ DDM searches for processes on the machines in your environment.
- ➤ DDM saves the process data (including open port and command line information) to the Probe database.
- ➤ The jobs run on this data in the Probe database and build the new Software Element CIs according to the data in the database (and extract the key attributes from the process data). The jobs send the CIs to the UCMDB server.

Running Jobs

Before running the Network – Host Resources module, you must run the following jobs. This is because these jobs are needed by the Process To Process by Network Data and Software Element CF by Shell jobs.

- ➤ Host Resource. To discover processes, activate Host Resources by SNMP, Host Resources by Shell, or Host Resources by WMI. For details, see "Network Host Resources" on page 315.
- ➤ **P2P Ports.** To discover open ports, process to port relationships, and TCP connections, activate **Collect Network Data By Shell or SNMP**. For details, see "Network Connections Active Discovery" on page 342.

Software Elements Default Views

Two default views exist that display the mapping of relationships between applications: Application Components and Application Components

Dependencies. To access the views: Admin > Universal CMDB > Modeling > View Manager > Root > Application > Application Signature.

You can configure DDM to discover software elements. For details, see "Discover Software Elements – Scenario" on page 200.

P Configure the DDM Probe to Automatically Delete CIs – Workflow

This task explains how to configure a job so that CI instances of specific CITs are automatically deleted. For details on how the DDM Probe deletes CIs, see "Automatically Deleted System Components" on page 196.

This task includes the following steps:

- ➤ "Prerequisites" on page 200
- ➤ "Select the CIs to be Deleted" on page 200
- ➤ "Results" on page 200

1 Prerequisites

Verify that the **Filter unchanged results** check box is selected in the Results Management pane in the Pattern Management tab. For details, see "Results Management Pane" on page 234.

2 Select the CIs to be Deleted

- **a** Select the **Automatically delete removed CIs** check box.
- **b** Click the **Add** button to open the Choose Discovered Class dialog box. For details, see "Choose Discovered Class Dialog Box" on page 210.
- **c** Click the **Save** button at the bottom of the page.

3 Results

To view the deleted CIs, access the Deleted column in the Statistics Results pane. For details, see "Statistics Results Pane" on page 117.

P Discover Software Elements – Scenario

This scenario explains how to set up the discovery of Oracle databases so that there is no need to enter a specific set of credentials to discover each database instance. DDM runs an extract command that retrieves the database name attribute

In this scenario, we assume that the following syntax is used in the Oracle command lines:

c:\ora10\bin\oracle.exe UCMDB

This task includes the following steps:

- ➤ "Prerequisites" on page 201
- ➤ "Create a Command Line Rule" on page 201
- ➤ "Define the Value of an Attribute" on page 202
- ➤ "Activate the Job" on page 203

1 Prerequisites

Display the Software Element Attribute Assignment Rules dialog box:

- a Select Admin > Universal CMDB > Discovery > Run Discovery. In the Discovery Modules pane, select Network Host Resources > Software Element CF by Shell. In the Properties tab, select Global Configuration Files > applicationSignature.xml. For details, see "Global Configuration Files Pane" on page 237.
- **b** Click the **Edit** button to open the Software Library dialog box. For details, see "Software Library Dialog Box" on page 248.
- c Select Software Element Categories > Database > Oracle by oracle.exe process. Click the Edit button to open the Software Identification Rule Editor dialog box. For details, see "Software Identification Rule Editor Dialog Box" on page 245.
- **d** Click the **Set Attributes** button to open the Software Element Attributes Assignment Rules dialog box. For details, see "Software Element Attribute Assignment Rules Dialog Box" on page 244.

2 Create a Command Line Rule

The command line rule is text that identifies the process to be discovered, for example, oracle.exe c:\ora10\bin\oracle.exe UCMDB. You can substitute the text entry with a regular expression, so that discovery is more flexible. For example, you can set up a rule that discovers all Oracle databases, whatever their name.

Subsequently, DDM uses the information in the command lines discovered by the regular expression to populate a CI's data_name attribute with the database name.

a To create a Command Line rule that includes a regular expression, in the Software Element Attributes Assignment Rules dialog box, click the **Add** button in the Parsing Rules pane. For details, see "Parse Rule Editor Dialog Box" on page 227.

- **b** In the Parse Rules Editor dialog box, build the rule:
 - ➤ Enter a unique name in the Rule ID field, for example, r1.
 - ➤ Choose **Command Line** in the Process Attribute field.
 - ➤ Enter the following regular expression in the Regular Expression field: .+\s+(\w+)\$:

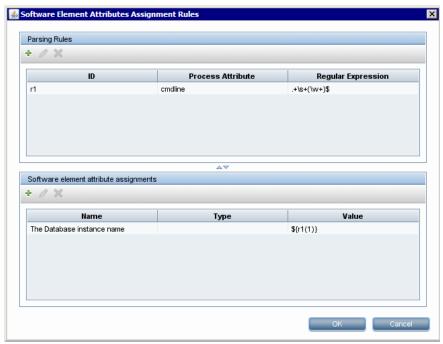
This expression searches for any character (.), followed by a space or spaces (+\s+), followed by a word or words ((\w+)) that appear at the end of the line (\$). The words can be a-z, A-Z, or 0-9. The following command line fulfils this expression: c:\ora10\bin\oracle.exe UCMDB.

3 Define the Value of an Attribute

In this step, you define which attribute is used by DDM to discover the Oracle databases, and the value it should take.

- **a** In the Software Element Attributes Assignment Rules dialog box, click the **Add** button in the Software Element Attribute Assignments pane, to select the attribute.
- **b** In the Attribute Editor dialog box:
 - ➤ Choose the attribute that holds the database name, from the list of Oracle CIT attributes, in this case **The Database instance name**.

➤ Enter a value, using the following syntax: \${<rule ID name>(<group number>)}, in this case, \${r1(1)}.



This dialog box is configured as follows: DDM should search for Oracle databases where the value of the database name attribute $({r1(1)})$ in the command line is equal to the first group $({w+})$ in the regular expression (for details, see step b on page 202).

That is, during discovery, DDM searches through the configuration files for command lines with a word or words at the end of the line. For example, the following command line matches this regular expression: c:\ora10\bin\oracle.exe UCMDB.

4 Activate the Job

For details, see "Manually Activate a Job" on page 52 and "Discovery Modules Pane" on page 119.

Resource Files

The following files can be changed to enable DDM in non-default systems. The location of these files is: **Manage Discovery Resources** > **Network** > **Configuration Files**.

This section includes the following topics:

- ➤ "portNumberToPortName.xml" on page 204
- ➤ "oidToHostClass.xml" on page 205
- ➤ "globalFiltering.xml" on page 206
- ➤ "Internal Configuration Files" on page 208

portNumberToPortName.xml

The portNumberToPortName.xml file is used by DDM as a dictionary to create Port CIs. When a port is discovered, the Probe extracts the port's name from this file, and creates the Port CI accordingly. If the port number does not appear in this file, the port name is used as the port number.

You edit this file when adding new ports to be discovered.

Note: The results of running a Network Connections – Active discovery appear in the Topology Map with the port names instead of the port numbers (the port title is the value of the Port Name attribute, defined in the CIT). For details, see "Add/Edit Attribute Dialog Box" in *Model Management*.

To define a new port:

1 In the Manage Discovery Resources window, access portNumberToPortName.xml and search for the file by clicking the Find resource button and entering portNumber in the Name box. Click Find Next, then click Close.

The file is selected in the Discovery Resources pane and the file contents are displayed in the View pane.

2 Add another row to the file and make changes to the parameters:

<portInfo portProtocol="xxx" portNumber="xxx" portName="xxx" discover="0"/>

- **portProtocol.** The network protocol used for discovery (udp or tcp).
- **> portNumber.** The port number to be discovered.
- **portName.** The name that is to be displayed for this port.
- ➤ **discover.** 1. This port must be discovered. 0: This port should not be discovered.

oidToHostClass.xml

The oidToHostClass.xml file contains a list of OID (Discovery and Dependency Mapping) numbers, for all CIs in the system that have an ID. This list is required for mapping CIs to their correct CIT, and for converting the discovered OID number of an operating system or a device into string data.

To access the oidToHostClass.xml file, in Manage Discovery Resources, search for the file by clicking the **Find resource** button and entering **oidto** in the **Name** box. Click **Find Next**, then click **Close**.

The file is selected in the Discovery Resources pane and the file contents are displayed in the View pane.

Note: If an OID is discovered and its details do not appear in the oidToHostClass.xml file, its CIT is registered in the CMDB as host.

oidToHostClass.xml includes the following parameters:

- ➤ class. The converted CIT name of the discovered OID. Under this name, the operating system or device appears in the CMDB and in Business Availability Center.
- ➤ vendor. The vendor of the operating system or device.

- ➤ **os.** A specific operating system, for example, Linux. This parameter is optional.
- ➤ model. A specific model, for example, JETDIRECT, JD30. This parameter is optional.
- ➤ **oid.** The discovered OID.

globalFiltering.xml

This file enables you to filter Probe results for all patterns, so that only results of interest to you are sent to the Business Availability Center server. (You can also filter specific patterns. For details, see "Pattern Management Tab" on page 228.)

To add a global filter:

- **1** Access the globalFiltering.xml file: in Manage Discovery Resources, open the Network folder and click the Configuration Files folder. Select the file to display the code in the View pane.
- **2** Locate the <includeFilter> and <excludeFilter> markers:
 - ➤ <includeFilter>. When a vector marker is added to this filter, all CIs that do not match the filter are removed. If this marker is left empty, all results are sent to the server.
 - ➤ <excludeFilter>. When a vector marker is added to this filter, all CIs that match the filter are removed. If this marker is left empty, all results are sent to the server.

The following example shows an ip CI that has address and domain attributes:

```
<vector>
    <object class="ip">
        <attribute name="ip_address" type="String">192\.168\.82\.17.*</attribute>
        <attribute name="ip_domain" type="String">DefaultProbe</attribute>
        </object>
</vector>
```

If this vector is defined in <includefilter>, all results not matching the filter are removed. The results sent to the server are those where the ip_address matches the regular expression 192\.168\.82\.17.* and the ip_domain is DefaultProbe.

If this vector is defined in <excludefilter>, all results matching the filter are removed. The results sent to the server are those where the ip_address does not match the regular expression 192\.168\.82\.17.* and the ip_domain is not DefaultProbe.

The following example shows a network CI that has no attributes. All network results are sent to the server:

```
<vector>
    <object class="network">
    </object>
</vector>
```

Note:

- ➤ Attributes in the filter should be of type string only. For details on attribute types, see "Attributes Page" in *Model Management*.
- ➤ A result is considered to be a match only if all filter attributes have the same values as those in the CI. (If one of a CI's attributes is not specified in the filter, all the results for this attribute match the filter.)
- ➤ A CI can match more than one filter. The CI is removed or remains according to the filter in which it is included.
- ➤ DDM filters first according to the <includeFilter> and then applies the <excludeFilter> on the results of <includeFilter>.

Internal Configuration Files

The following files are for internal use only and should be changed only by users with an advanced knowledge of pattern-writing:

- ➤ discoveryPolicy.xml. Includes the schedule when the Probe does not execute tasks. For details, see "Add/Edit Policy Dialog Box" on page 166.
- ➤ jythonGlobalLibs.xml. A list of default Jython global libraries that DDM loads before running scripts.

🍳 Manage Discovery Resources User Interface

This section describes:

- ➤ Attribute Editor Dialog Box on page 209
- ➤ Choose Discovered Class Dialog Box on page 210
- ➤ Configuration File Pane on page 212
- ➤ Discovery Pattern Source Editor Window on page 213
- ➤ Discovery Resources Pane on page 215
- ➤ Find Discovery Resource Dialog Box on page 218
- ➤ Find Text Dialog Box on page 220
- ➤ Input TQL Editor Window on page 220
- ➤ Manage Discovery Resources Window on page 224
- ➤ Parse Rule Editor Dialog Box on page 227
- ➤ Pattern Management Tab on page 228
- ➤ Pattern Signature Tab on page 235
- ➤ Permission Editor Dialog Box on page 241
- ➤ Script Editor Window on page 242
- ➤ Script Pane on page 243
- ➤ Software Element Attribute Assignment Rules Dialog Box on page 244
- ➤ Software Identification Rule Editor Dialog Box on page 245

➤ Software Library Dialog Box on page 248

Natural Editor Dialog Box

Description	Enables you to define a rule that discovers a CIT according to an attribute. The attribute is defined according to a regular expression.
	To access: Click the Add button in the Software Element Additional Attributes pane in the Software Element Attributes Editor dialog box.
Included in Tasks	"Discover Software Elements – Scenario" on page 200
Useful Links	"Parse Rule Editor Dialog Box" on page 227

GUI Element (A–Z)	Description	
Name	Choose from the list of attributes of the CIT selected in the Software Element Editor. This attribute name is replaced by the value found by the regular expression.	
Туре	The type of operation defined for the attribute, for example, Boolean, string, date, and so on.	
Value	The value that replaces the name in the Rule ID field in the Parse Rule Editor dialog box.	
	Use the following syntax for the value:	
	\${ <rule id="" name="">(<group number="">)}</group></rule>	
	For example, \${DB_SID(1)} means that DDM should search for the Rule ID with the name DB_SID and retrieve its regular expression.	
	DDM should then retrieve the code for the first group (1). For example, in the regular expression $.+\s+(\w+)$ \$, the first group is $(\w+)$ \$, that is, a word or words that appear at the end of the line.	

Choose Discovered Class Dialog Box

Description	Enables you to choose CITs that are to be discovered by a selected pattern and to limit links so that they are mapped only when they connect specific CITs.
	To access:
	 Admin > Universal CMDB > Discovery > Manage Discovery Resources. In the Discovery Resources pane, select a pattern. In the Pattern Signature tab > Discovered CITs pane, click the Add Discovered CIT button. Admin > Universal CMDB > Discovery > Manage Discovery Resources. In the Discovery Resources pane, select a pattern. In the Pattern Management tab > Automatic Deletion pane, select the Enable Automatic deletion of removed CIs check box and click the Add button.

GUI Element (A–Z)	Description
Link	Select a link type from the list and click the button in the End 1 and End 2 boxes to open the Choose Configuration Item Type dialog box. Choose the CITs that DDM should map when they are linked by the selected link type.
	Note: DDM automatically recognizes the links between CIs and adds them to the map of discovered CITs. However, during pattern writing, you may need to exclude links between certain CITs. For example, both hosts and IPs and hosts and ports are linked by use. You may need to receive results only for hosts and IPs that are connected by the use link, and not hosts and ports. The End 1 and End 2 links determine the result received from the pattern, and this result is reflected in the map, as can be seen in the following example:
	Web Server Virtual Host use container_f container_f container_f container_f Configuration File container_f container_f
Object	Select a CIT to be added to the list of CITs that a pattern is to discover. Save the changes by clicking the Save
	button at the bottom of the Pattern Signature pane.

Configuration File Pane

Description	Enables you to edit a specific configuration file that is part of a package. For example, you can edit the portNumberToPortName.xml file so that specific port numbers, names, or types are discovered.
	To access: Click a specific configuration file in the Discovery Resources pane.
Important Information	The following files are for internal use only and should only be changed by users with an advanced knowledge of pattern-writing:
	➤ discoveryPolicy.xml
	➤ jythonGlobalLibs.xml
	For details, see "Resource Files" on page 204 and "Internal Configuration Files" on page 208.

GUI Element (A–Z)	Description
Q	Find specific text in the configuration file. For details, see "Find Text Dialog Box" on page 220.
9	Click to go to a specific line in the configuration file. In the Go To Line dialog box, enter the line number.
	Click to open the file in an external editor.

GUI Element (A–Z)	Description
827	Click to edit the external editor preferences. You can run the editor by adding flags to the path.
	In the following example:
	Select External editor path Full Path C:\anyTextEditor.exe Flags -I-k:file -v Cancel :file sets the place of the file in relation to the flags. The user cannot set the file name.
XML	For XML files, signifies that the code is valid.
XML	For XML files, signifies that the code is not valid.

Discovery Pattern Source Editor Window

Description	Enables you to edit a pattern script.
	To access: Right-click a pattern in the Discovery Resources pane and select Edit Pattern Source .

GUI Element (A–Z)	Description
Q	Find specific text in the pattern script. For details, see "Find Text Dialog Box" on page 220.
9	Click to go to a specific line in the pattern script. In the Go To Line dialog box, enter the line number.
	Click to open the pattern script in an external text editor.

Chapter 6 • Manage Discovery Resources

GUI Element (A–Z)	Description
82	Click to edit the external editor preferences. You can run the editor by adding flags to the path.
	In the following example:
	Select External editor path Full Path C:\tanyTextEditor.exe Flags -I -k: file -v Sile sets the place of the file in relation to the flags. The user cannot set the file name.
XML	Signifies that the code is valid.
XML	Signifies that the code is invalid.

Discovery Resources Pane

Description	Enables you to locate a specific package, pattern, script, configuration file, or external resource. You can also create a pattern, Jython script, configuration file, or Discovery wizard, and you can import an external resource. To access: Admin > Universal CMDB > Discovery > Manage Discovery Resources
Important Information	Depending which level you select in the Discovery Resources pane, different information is displayed in the View pane. If you select: One of the following folders: Discovery Packages root, a specific package, a pattern, script, configuration file, or external resource: a list of the resources in that folder is displayed. To access a resource directly, double-click the resource in the View pane. A specific pattern: The Pattern Signature and Pattern Management panes are displayed. For details, see "Pattern Signature Tab" on page 235 and "Pattern Management Tab" on page 228. A script or configuration file: The script editor is displayed. For details, see "Script Pane" on page 243. An external resource: Information about the file is
Useful Links	displayed. "Package Manager User Interface" in <i>Model Management</i> .

Chapter 6 • Manage Discovery Resources

GUI Element (A–Z)	Description
*	Click to:
	➤ Create a pattern. Enter the pattern name and click OK. The new pattern is added to the << No Package >> folder. Edit the pattern. For details, see "Pattern Signature Tab" on page 235 and "Pattern Management Tab" on page 228. For details on moving a pattern to a package, see "Create New Package/Edit Package Wizard" in Model Management.
	➤ Create a Jython script. Enter the script name. For details, see "Script Pane" on page 243.
	➤ Create a configuration file. Enter the configuration file name. By default, the file takes an .xml extension. To give the file another extension, for example, *.properties, name the file and include the extension. Add the appropriate XML code or other content. For XML files, you can save the file only if it is valid. For details, see "Configuration File Pane" on page 212.
	➤ Import an external resource. In the browser that opens, locate the resource to be imported and click Open .
	➤ Create a Discovery Wizard. Name the new wizard. By default, the file takes an .xml extension. A new file is added to the Discovery Wizard folder of the << No Package >> folder. The file is in template format.
×	Click to delete the resource.
0	Click to refresh the list of packages.
Q	Click to open the Find Discovery Resource dialog box. For details on filtering, see "Filtering Results" on page 35.
#	Discovery packages root. Displays a list of all packages.
<u> </u>	Package root. Displays a list of all resources included in the package. You can view any of these resources by clicking the resource in the Discovery Resources pane.

GUI Element (A–Z)	Description
<script files=""></th><th>Right-click a file to:</th></tr><tr><th></th><th> Save as. Save the file under a new name. Use this option to clone an existing file. The new file includes all attributes of the existing file. Make any necessary changes to the file and save it. Open in Frame. Select to open the file in a new window. </th></tr><tr><th><Configuration</th><th>Right-click a file to:</th></tr><tr><th>files></th><th> Save as. Save the file under a new name. Use this option to clone an existing file. The new file includes all attributes of the existing file. Make any necessary changes to the file and save it. Open in Frame. Select to open the file in a new window. </th></tr><tr><th><External resource files></th><th>An external resource is any file needed by DDM to perform discovery. For example, the nmap.exe file is needed for credential-less discovery.</th></tr><tr><th></th><th>➤ Right-click a file to:</th></tr><tr><th></th><th> ➤ Save as. Save the resource under a new name. Use this option to clone an existing resource. The new resource includes all attributes of the existing resource and is saved to the same location in the file system. Make any necessary changes to the new resource and save it. ➤ Select the file to display information in the View pane. You can open an external resource or export it. </th></tr></tbody></table></script>	

GUI Element (A–Z)	Description
<pattern files=""></pattern>	Right-click a file to:
	 Save as. Save the pattern under a new name. Use this option to clone an existing pattern. The new pattern includes all attributes of the existing pattern. Give a name to the new pattern, and change the necessary attributes. Go to Discovery job. When enabled, click to open the Run Discovery window with the job selected.
	This option is enabled if the pattern is included in a job.
	➤ Edit pattern source. Opens the pattern source editor where you can make changes to the pattern. For details, see "Discovery Pattern Source Editor Window" on page 213.
<script files=""></th><th>Right-click a file to:</th></tr><tr><th></th><th> Save as. Save the script under a new name. Use this option to clone an existing script. The new script includes all attributes of the existing script. Make any necessary changes to the script and save it. Open in Frame. Select to open the script in a new </th></tr><tr><th></th><th>window. For details on editing the script, see "Discovery Pattern Source Editor Window" on page 213.</th></tr></tbody></table></script>	

Find Discovery Resource Dialog Box

Description	Enables you to build a search query to find a particular resource.
	To access: Click the Find Job filter button in the Discovery Resources pane.

GUI Element (A–Z)	Description
	Click to select a CIT from the dialog box that opens. Click OK to return to the Find Discovery Resource dialog box.
	Note: This button is not accessible when Name is selected.
Direction	Searches forwards or backwards through the packages.
Find All	All resources meeting the search criteria are highlighted in the Discovery Resources pane.
Find Discovery resource by	Choose between:
	➤ Name. Enter the name, or part of it, of the resources.
	➤ Pattern input type. CIs that trigger the job. Click the button to open the Choose Configuration Item Type dialog box. Locate the CI type that you are searching for.
	➤ Pattern output type. CIs that are discovered as a result of the activated job.
Find Next	The next resource meeting the search criteria is highlighted in the Discovery Resources pane.

🙎 Find Text Dialog Box

Description	Enables you to find text in a script or configuration file.
	To access: Click the Find in text button.

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Direction	Search up or down the script.
Options	Select Case Sensitive to narrow your search.
Text to find	Enter the text you are searching for.

🔍 Input TQL Editor Window

Description	Enables you to define which CIs can be Trigger CIs for jobs that run a specific pattern.
	To access: Manage Discovery Resources > select a pattern > Pattern Signature tab > click the Edit button next to the Input TQL box.
Useful Links	 ➤ "Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs" on page 50 ➤ "Trigger TQL Editor Window" on page 154

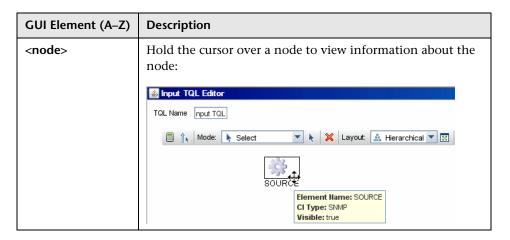
GUI Element (A–Z)	Description
<panes></panes>	➤ CI Type Selector Pane
	➤ Editing Pane
	➤ Information Pane
TQL Name	The name of the trigger TQL query that activates the job.

CI Type Selector Pane

Description	Displays a hierarchical tree structure of the CI Types found in the CMDB. For more details, see "CI Type Manager User Interface" in <i>Model Management</i> .
	Note: The number of instances of each CIT in the CMDB is displayed to the right of each CIT.
	To create or modify a TQL query, click and drag nodes to the Editing pane and define the relationship between them. Your changes are saved to the CMDB. For details, see "Add Nodes and Relationships to a TQL Query" in <i>Model Management</i> .
Included in Tasks	 ➤ "Define a TQL Query" on page 504 in Model Management ➤ "Create a Pattern View" in Model Management

Editing Pane

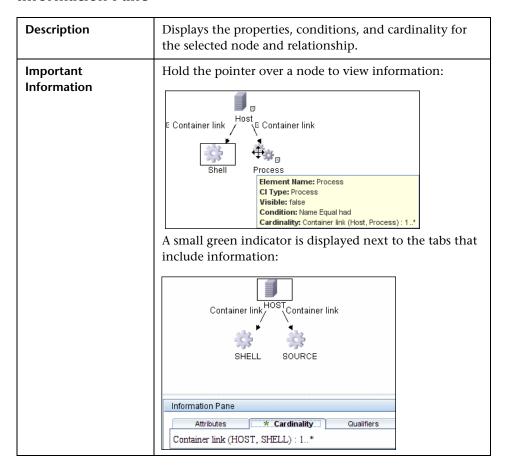
Description	Enables you to edit the node.
-------------	-------------------------------



Chapter 6 • Manage Discovery Resources

GUI Element (A–Z)	Description
<right-click menu=""></right-click>	For details, see "Context Menu Options" in <i>Model Management</i>
<toolbar></toolbar>	For details, see "Toolbar Options" in Model Management

Information Pane



GUI Element (A-Z)	Description
⊘	Click Edit to open the relevant dialog box for the selected tab.
Attributes	Displays the attribute conditions defined for the node or the relationship. For details, see "Attribute Tab" in <i>Model Management</i> .
Cardinality	Cardinality defines how many nodes you expect to have at the other end of a relationship. For example, in a relationship between host and IP, if the cardinality is 1:3, the TQL retrieves only those hosts that are connected to between one and three IPs. For details, see "Cardinality Tab" in <i>Model Management</i> .
Details	 ➤ To open the Node or Relationship Properties dialog box, select a node or relationship in the Editing pane and click the Edit button. For details, see "Node/Relationship Properties Dialog Box" in Model Management. ➤ CI Type. The CIT of the selected node/relationship. ➤ Visible. A tick signifies that the selected node/relationship is visible in the topology map. When the node/relationship is not visible, a box is displayed to the right of the selected node/relationship in the Editing pane: ➤ Include subtypes. Display both the selected CI and its descendants in the topology map.
Qualifiers	Displays the qualifier conditions defined for the node or the relationship. For details, see "Qualifier Tab" on page 565.

GUI Element (A-Z)	Description
Selected Identities	Displays the element instances that are used to define what should be included in the TQL results. For details, see "Identity Tab" on page 567.

Manage Discovery Resources Window

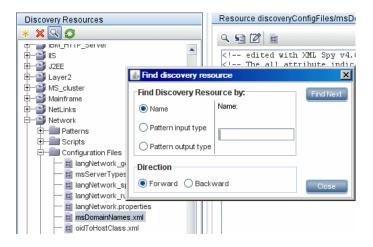
Description	Enables you to view or edit default parameter values used for the DDM process. To access: Admin > Universal CMDB > Discovery > Manage Discovery Resources or right-click a job in the Run Discovery window.
Important Information	Note: An asterisk (*) next to a resource (pattern, script, or configuration file) signifies that the resource has changed since the package (in which it is included) was deployed. If the original package is redeployed, the changes are deleted from the resource. To save the changes, move the resource to a new package and deploy the package (the asterisk disappears). Caution: Only administrators with an expert knowledge of the DDM process should delete packages.
Useful Links	"Pattern Signature Tab" on page 235 "Global Configuration Files Pane" on page 237 "Pattern Management Tab" on page 228 "Script Pane" on page 243 "Discovery Resources Pane" on page 215 "Configuration File Pane" on page 212 Chapter 8, "Discovery and Dependency Mapping Content"

Important: The following examples include steps that modify resource files. Modification is considered a more advanced configuration and should be performed by users with an advanced knowledge of Business Availability Center only, for site-specific customization.

Example - Set Up Discovery of Microsoft Domains

To discover specific Microsoft domains on a network:

1 In the Manage Discovery Resources window, search for the msDomainNames.xml file by clicking the **Find resource** button and entering **domain** in the **Name** box:



DDM highlights the resource that includes this string.

2 Click **Find Next** until DDM highlights msDomainNames.xml in the Network package. Click **Close**.

DDM displays the file in the View pane with the results of the XML validation.

3 Locate the **<MsDomainNames all="true">** marker and change **true** to **false**.

false: DDM finds only those Microsoft domain types specified in the <MsDomain> list.

true: DDM finds all existing Microsoft domain types and not only the types listed in <MsDomain>.

4 Locate the **<MsDomain>** marker and replace DISCOVERED-MS-DOMAIN with the name of the Microsoft domain.

(To add more domains to be discovered, add a <MsDomain>Domain to be discovered<MsDomain> marker for each domain.)

Example:

```
<MsDomainNames all="false">
    <MsDomain>Business Availability Center Lab</MsDomain>
    <MsDomain>Finance_servers</MsDomain>
</MsDomainNames>
```

5 Save the file. Verify that the file is valid XML by checking the Validation info.

Example - Set Up Discovery of Novell Servers

To discover the host attributes of a Novell server on a network:

- 1 In the Manage Discovery Resources window, search for the msServerTypes.xml file by clicking the Find resource button and entering msser in the Name box:
- **2** Click **Find Next** until DDM highlights msServerTypes.xml in the Network package. Click **Close**.
 - DDM displays the file in the View pane with the results of the XML validation.
- **3** Locate the <MsServerType value="80" name="NOVELL" calc="false"/> marker and change false to true.

false: DDM does not save data about this server to the database.

true: DDM saves the CIs to the CMDB (as attributes of their host).

(**value** is the discovered number of the server type. **name** is the converted name of the discovered server type. The server type appears under this name in the CMDB and in Business Availability Center.)

Example – Configure Resources for a New Package

To define a package for a pattern and configuration file that you have written for a non-default discovery:

1 To create a pattern, you can either define a new pattern or copy an existing pattern.



To define a new pattern, in the Manage Discovery Resources window, click the icon in the Discovery Resources toolbar and select **New Discovery Pattern**. For details, see "Discovery Resources Pane" on page 215.

To copy an existing pattern, select the pattern, and save the file under a new name. The new pattern is located under the << No Packages >> folder.

- **2** Define the configuration file in the same way, either by defining a new file or by copying an existing file.
- **3** In Package Manager, define a new package and add the new resources to the package. For details, see "Package Manager User Interface" in *Model Management*.

💐 Parse Rule Editor Dialog Box

Description	Enables you to create a rule that matches an attribute to process-related information (IP, port, command line, and owner).
	To access: Click Set Attributes in the Software Element Editor dialog box.
Important Information	Only users with a knowledge of regular expressions should make changes to a rule.
Included in Tasks	"Discover Software Elements – Scenario" on page 200
Useful Links	"Attribute Editor Dialog Box" on page 209

Chapter 6 • Manage Discovery Resources

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Rule ID	Enter a unique name for the rule. The Rule ID is needed to identify the rule in the Software Element Additional Attributes value.
Process Attribute	Choose between the Port , IP , Command Line , or Owner process-related information. The rule is invoked on the attribute you choose here.
Regular Expression	Enables you to create a dynamic expression that finds at least one process that defines this software element. The regular expression is invoked on the value in the Process Attribute field.
	For example, a command line process includes the following regular expression:
	.+\s+(\w+)\$
	This expression searches for any character, followed by a space or spaces, followed by a word or words (a-z or A-Z or 0-9) that appear at the end of the line.
	The following command line matches this regular expression: c:\ora10\bin\oracle.exe UCMDB

😢 Pattern Management Tab

Description	Enables you to define how DDM filters the results.
	To access: Select a specific pattern in the Discovery Resources pane.
Important Information	Click the Save button to save any changes you make.
Useful Links	"The DiscoveryProbe.properties File" on page 38

Automatic Deletion Pane

GUI Element (A–Z)	Description
Enable automatic deletion of removed CIs	Select this check box to automatically delete CIs that should be removed from the database if the DDM Probe does not find them during its next invocation.
	To add CITs to the list of CIs, click the Add button. In the Choose Configuration Item Type dialog box, choose the CITs that should be automatically deleted.
	The changes you make here are added to the pattern file, for example:
	<pre><resultmechanism isenabled="true"> <autodeletecits isenabled="true"></autodeletecits></resultmechanism></pre>
	For details on how the DDM Probe handles CI deletion, see "Automatically Deleted System Components" on page 196.
	Note: This check box is enabled only when the Filter unchanged results check box in the Results Management pane is selected.

Execution Options Pane

GUI Element (A–Z)	Description
Create communication logs	Choose to create a log file that logs the connection between the Probe and a remote machine. For details on how the communication logs work, see "Record DDM Code" on page 436.
	➤ Always/On Failure. When selected, an ad-hoc request is set up to retrieve the communication log for the selected Trigger CI.
	Always. DDM always creates communication logs.
	On Failure. DDM automatically creates a log of the session in case discovery should fail (that is, DDM throws an exception or reports an error; report of a warning does not create a communication log). This is useful when you need to analyze which queries or operations take most of the time, send data for analysis from different locations, and so on. If the job completes successfully, no log is created.
	➤ Never. No communication logs are saved to the file system or to memory. The communication log can reside in two places: in the Probe's memory or on the file system (as a record file). When requested, DDM displays the log retrieved from the Probe's memory. If no log exists in the memory, DDM displays the log that resides on the file system. If no log exists on the file system, a message is displayed. The communication log files are created on the Probe Manager under the C:\hp\DDM\DiscoveryProbe \root\lib\collectors\probeManager\record folder.

GUI Element (A–Z)	Description
Enable Automatic Deletion	Choose between:
	➤ Always. Automatic Deletion is always enabled, regardless of the discovery success status indication.
	➤ On Success or Warnings. Automatic Deletion is enabled only when discovery finishes with a success or warning status. In the case of a discovery error, nothing is removed.
	➤ Only on Success. Automatic Deletion is enabled only when discovery finishes with a success status. In the case of a discovery error or warning, nothing is removed (this is the default).
	This element functions together with the Enable automatic deletion of removed CIs check box in the Automatic Deletion pane.
	For details on discovery status, see "Discovery Status Pane" on page 110.
Include Results in Communication Log	Select to enable capturing the discovered results with the created communication log; these discovered results may help in investigating various discovery problems.
Max. Execution Time	The maximum time allowed for a pattern to run on one Trigger CI.
Max. Threads	Each job is run using multiple threads. You can define a maximum number of threads that can be used concurrently when running a job. If you leave this box empty, the Probe's default threading value is used (8).
	The default value is defined in DiscoveryProbe.properties in the defaultMaxJobThreads parameter.
	➤ regularPoolThreads. The maximum number of worker threads allocated to the multi-threaded activity (the default is 50).
	➤ priorityPoolThreads. The maximum number of priority worker threads (the default is 20).
	Note: The number of actual threads should never be higher than regularPoolThreads + priorityPoolThreads.

General Options Pane

The following elements are included:

 ▶ Selected. DDM collects data on the respectation. This data is then used to enable CIs. The data is necessary for the Discount Universe to function correctly. It is also Based Discovery Status functionality we data to aggregate the complete discovery views. ▶ Cleared. DDM does not collect this data needs to be cleared for patterns where helpful. For example, the Range IPs by check box cleared by default because it Probe Gateway and so all CIs discovered the same Trigger CI. If the check box we rediscovery attempt on any view contawould result in a ping sweep throughout customer network, certainly not desira. The job results of this pattern are displayed for View dialog box only if this check box details, see "Check Status of Application In (Rediscover a View)" and "Discovery and Dialog Box" in Model Management. 	le rediscovery of overy tab in IT o used for the View which leverages the ry status for certain ta. The check box rediscovery is not o ICMP job has this to Trigger CI is the ed by this job have was not cleared, a lining any single IP out the entire ble behavior. ed in the Discovery of is selected. For Discovery

Probe Selection Pane

Description	Enables you to specify which Probe to use with a pattern.
	To access: Select a specific pattern in the Discovery Resources pane.
Important Information	By default, DDM automatically chooses the Probe for the Trigger CI according to the CI's related host. After obtaining the CI's related host, DDM chooses one of the host's IPs and chooses the Probe according to the Probe's network scope definitions.
	This may fail in the following situations:
	➤ A Trigger CI does not have a related host (such as the network CIT).
	➤ A triggered CI's host has multiple IPs, each belonging to a different Probe.
	To resolve these issues, you can specify which Probe to use with the pattern by:
	➤ In the Probe Selection section, selecting Override default probe selection.
	➤ In the Probe box, typing the Probe to use for the task.

GUI Element (A–Z)	Description
Override default	You can use calculated values such as:
probe selection	\${Network.network_domain}
	This value uses a syntax similar to that used by Triggered CI data in the Pattern Signature tab. For details, see "Triggered CI Data Pane" on page 239.

Result Grouping Pane

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
Grouping Interval (Seconds)	To group results in the Probe before they are sent to the server, type the value that indicates how long results are stored in the Probe before being transferred to the server. Note: If you enter a value in both boxes, DDM applies the value of whichever occurs first.
Max. Cls in group	Specify the number of CIs that should accumulate in the Probe before being transferred to the server.

Results Management Pane

GUI Element (A–Z)	Description
Enable aging	Select this check box to run the aging mechanism that specifies how long a period must pass in which CIs are discovered, before DDM treats these CIs as no longer relevant and removes them.
	Aging parameters are defined in the Infrastructure Settings Manager (Admin > Platform > Setup and Maintenance > Infrastructure Settings):
	➤ Aging Scheduler Hour of the First Run. Defines at what time aging first runs after server startup (for example, 02=2 AM).
	➤ Aging Scheduler Interval. Defines the interval between runs. If Aging Time Unit = days, the interval value is days; if Aging Time Unit = hours, the interval value is hours.
	➤ Aging Time Unit. The default is days . The hours option is provided to enable convenient verification checks of specific CIs.

GUI Element (A–Z)	Description
Filter unchanged results	Select this check box for the Probe to send to the CMDB only those CIs that are unchanged since the last time results were sent to the server and that answer to the filter criteria. For details on filtering, see "Filtering Results" on page 35.

🙎 Pattern Signature Tab

Description	Enables you to define a pattern by specifying:
	➤ which CITs the pattern should discover
	➤ which protocols are needed to perform discovery
	To access: Select a specific pattern in the Discovery
	Resources pane.
Included in Tasks	"Implement a Pattern" on page 411

GUI Element (A–Z)	Description
	Click the button to open the Input TQL editor. For details, see "Input TQL Editor Window" on page 220.
×	Click to remove the Input TQL from the pattern.
Description	The description of the pattern.
Input TQL	Defines which CIs can be Trigger CIs for jobs that run this pattern. Note: Since this field is optional, not all patterns include an input TQL. NA signifies not applicable, that is, currently this pattern does not have an input TQL definition. Click the button to open the Input TQL Editor window. For details, see "Input TQL Editor Window" on page 220. For an explanation, see "Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs" on page 50. For an example, see "Example of Input TQL Definition" on page 413.

GUI Element (A–Z)	Description
Trigger CIT	The CIT used as the trigger that activates the selected pattern. The trigger CIT is used as the pattern input. For details, see "Define Pattern Input (Trigger CIT and Input TQL)" on page 412. Click the button to choose a CIT to use as the trigger.

Discovered CITs Pane

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
+	Click to open the Choose Discovered Class dialog box, to select a CIT that is to be discovered by the pattern. For details, see "Choose Discovered Class Dialog Box" on page 210.
×	Click to remove the CIT from the list of CITs that the pattern discovers.
	You can choose to view a map of the CITs and links that are discovered by the pattern, instead of a list. Click the button to open the Discovered CITs Map window. The CIs and relationship links discovered by the pattern are shown.
CITs	List of CITs that the pattern discovers.

Discovery Pattern Parameters Pane

GUI Element (A–Z)	Description
÷	Click to open the Parameter Editor. Enter details on the parameter. The value you enter here is assigned to the attribute.
	Select a parameter and click the button to open the Parameter Editor and make changes.

GUI Element (A–Z)	Description
×	Click to remove a parameter.
Name	Each row represents the definitions for one parameter.
Value	Separate values with commas.

Global Configuration Files Pane

Description	Enables you to add default configuration files to the pattern, as well as the specific configuration files that the pattern needs.
	To access:
	 In Manage Resources, select a pattern and the Pattern Signature tab. In Run Discovery, select a job and the Properties tab.
	In Run Discovery, select a job and the Properties tab.
Important Information	The configuration file applicationsSignature.xml opens the Software Library dialog box. For details, see "Software Library Dialog Box" on page 248.
	The applicationsSignature.xml file contains a list of all applications that DDM attempts to find in the environment.
Included in Tasks	"Discover Software Elements – Scenario" on page 200

GUI Element (A–Z)	Description
+	Click to open the Global Configuration Files dialog box, to select configuration files that are needed by the pattern.
×	Click to delete a selected configuration file.
⊘	Select a configuration file and click to open the appropriate editor. For example, the file msServerTypes.xml opens the Script Editor.

Required Permissions Pane

Description	Enables you to view the permissions that you have configured for a pattern.
	To access: Manage Discovery Resources > select a pattern > Pattern Signature tab > Required Permissions pane.
Important Information	 Workflow: Configure the permissions in the Permission Editor dialog box. View the permissions in this pane. When working with jobs in the Run Discovery window, view these permissions for a specific job. For details on the fields in this pane, see "Permission Editor Dialog Box" on page 241.
Useful Links	 ➤ "Permission Editor Dialog Box" on page 241 ➤ "Discovery Permissions Window" on page 123 ➤ "Viewing Permissions While Running Jobs" on page 71

GUI Element (A–Z)	Description
+	Click to add a permission object. The Permission Editor dialog box opens. For details, see "Permission Editor Dialog Box" on page 241.
×	Select a permission object and click to delete it.
	Select a permission object and click the button to edit. For details, see "Permission Editor Dialog Box" on page 241.
1	Change the order of the permissions by selecting the permission object and clicking the up or down button. The order given here is the order in which the credentials are verified.
	Export a permission object in Excel, PDF, RFT, CSV, or XML format. For details, see "Browse Mode" in <i>Model Management</i> .

Required Discovery Protocols Pane

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
+	Opens the Add Required Protocol dialog box.
×	Click to remove an existing protocol.
Protocols	List of protocols required by the pattern for the task. For example, the NTCmd protocol, together with its user name, password, and other parameters, is needed for DDM to access a Windows system.

Triggered CI Data Pane

GUI Element (A–Z)	Description
+	Add Trigger CI data to the pattern.
×	Remove Trigger CI data from the pattern.
②	Edit the Trigger CI data in the Parameter Editor dialog box.

GUI Element (A–Z)	Description
Name	The information that is needed to perform a task on a specific CI. This information is passed to the CI queried in the task.
Value	The attribute value. Variables are written using the following syntax:
	\${VARIABLE_NAME.attributeName}
	where VARIABLE_NAME can be one of three predefined variables:
	➤ SOURCE . The CI that functions as the task's trigger.
	➤ HOST . The host in which the triggered CI is contained.
	➤ PARAMETERS. The parameter defined in the Parameter section.
	You can create a variable. For example, \${SOURCE.network_netaddr} indicates that the trigger CI is a network.

Used Scripts Pane

GUI Element (A–Z)	Description
₩ ₩	Change the order of the scripts. DDM runs the scripts in the order in which they appear here.
÷	Add a script to the pattern.
×	Remove a script from the pattern.
Ø	Edit the selected script in the Script Editor that opens.
Scripts	A list of Jython scripts used by the pattern. The Jython scripts that appear in bold are the scripts that the currently selected pattern is using.

Permission Editor Dialog Box

Description	Enables you to configure a pattern you have written, so that users can view permissions for the job.
	To access: Manage Discovery Resources > select a pattern > Pattern Signature tab > Required Permissions pane > click the Add button.
Important Information	The information you define here is not dynamic, that is, if a pattern is changed, the information in this dialog box is not updated.
Useful Links	 "Discovery Permissions Window" on page 123 "Viewing Permissions While Running Jobs" on page 71 "Required Permissions Pane" on page 238 "Discovery Job Details Pane" on page 109

GUI Element (A–Z)	Description
Permission	Enter a name for the permission, to appear in the Required Permissions pane.
Operation	The action that is being run.
Usage Description	Free text that you enter to describe the permission object and its parameters. This text is usually a general comment on the type of permission object, whereas the description is a more specific comment. For example, you could enter Permissions for host machines here Permissions for host machines running on Windows for a particular row.

Permission Objects and Parameters Dialog Box

GUI Element (A–Z)	Description
÷	Click to open the Permission Object and Parameter dialog box. You can enter more than one object or parameter for each permission.
	The information you enter in this dialog box appears in the Required Permissions pane, in the Objects and Parameters column.
×	Click to delete a permission object.
⊘	Click to edit an existing permission object.
Context	Specific information about the permission object's environment, for example, Windows or UNIX.
Parameter	The parameters that are needed during the job run. For example, the UNIX permission object cat needs the /etc/passwd parameter.
Permission Object	The name of the command, table, or other content of the Jython script.

Script Editor Window

Description	Enables you to edit a specific script that is part of a package.
	To access:
	➤ Right-click a script in the Discovery Resources pane and choose Open in Frame .
	➤ Select a configuration file in the Global Configuration Files pane and click the Edit button. For details, see "Script Pane" on page 243.

% Script Pane

Description	Enables you to edit a specific script that is part of a package. To access: Click a specific script in the Discovery Resources pane.
Important Information	The script pane title bar includes the actual physical location of the script. For example, the following script is located in C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeMa nager\discoveryScripts (or probeGateway \discoveryScripts): Resource discoveryScripts/SNMP_Dis_F5BIGIP_LTM.py Q

GUI Element (A–Z)	Description
Q	Find specific text in the script. For details, see "Find Text Dialog Box" on page 220.
9	Click to go to a specific line in a script. In the Go To Line dialog box, enter the line number.
	Click to open the script in an external text editor.
::1	Click to edit the external editor preferences. You can run the editor by adding flags to the path. In the following example: Select External editor path Full Path C:\anyTextEditor.exe Flags -1-k:file-v :file sets the place of the file in relation to the flags. The user
	cannot set the file name.

GUI Element (A–Z)	Description
XML	For Jython files, signifies that the code is valid.
XML	For Jython files, signifies that the code is not valid.
Fix validation errors	See Validation Information below.
<script></th><th>The Jython script used by the package. For details on working with Jython, see "Step 3: Create Jython Code" on page 422.</th></tr><tr><th>Validation Information</th><th>If a script is not valid in version 8.00, Validation Information displays the errors in the script, for example:</th></tr><tr><th></th><th>Script has failed validation. At line 48: Factory.getProtocolProperty(found. This is a problem - Usage of Factory is deprecated. Use Framework.getProtocolProperty instead.</th></tr><tr><th></th><th>Click Fix validation errors then OK to update the script.</th></tr><tr><th></th><th>The error may occur due to changes in the Framework object's API.</th></tr></tbody></table></script>	

Software Element Attribute Assignment Rules Dialog Box

Description	Enables you to define a regular expression that discovers a specific software element according to a CIT's attribute value.
	To access: Click Set Attributes in the Software Element Editor dialog box.
Included in Tasks	"Discover Software Elements – Scenario" on page 200
Useful Links	 "Parse Rule Editor Dialog Box" on page 227 "Attribute Editor Dialog Box" on page 209

The following elements are included (unlabeled GUI elements are shown in angle brackets):

GUI Element (A–Z)	Description
+	Click to add a regular expression that determines the attribute of the CI to be discovered, or to add an attribute.
×	Click to delete the regular expression or the attribute.
⊘	Click to edit an existing regular expression or attribute.
Parse Rules	For details, see "Parse Rule Editor Dialog Box" on page 227.
Software Element Additional Attributes	For details, see "Attribute Editor Dialog Box" on page 209.

🔍 Software Identification Rule Editor Dialog Box

Description	Enables you to define a new software element.
	To access: In the Software Library dialog box, click the Add button or select an existing element and click the Edit button.
Included in Tasks	"Discover Software Elements – Scenario" on page 200
Useful Links	"Global Configuration Files Pane" on page 237

GUI Element (A–Z)	Description
Set Attributes	Click to add attributes to the component. For details, see "Software Element Attribute Assignment Rules Dialog Box" on page 244.
÷	Click to add a process or a software configuration file.

Chapter 6 • Manage Discovery Resources

GUI Element (A–Z)	Description
×	Select a process or a software configuration file and click to delete.
	Select a process or a software configuration file and click to edit.
Category	You can:
	➤ Choose the category under which the new software element should appear.
	➤ Change the category for an existing element.
	➤ Add a new category by typing its name in this field.
	The changes you make here are immediately displayed in the Software Library dialog box.
CI Type	Select the CIT that is to be discovered.
Identifying Configuration Files	To add a configuration file, click the Add button. The Configuration File Name dialog box opens. Enter the full path to the software element's configuration file and the file name.

GUI Element (A–Z)	Description
Identifying Processes	To add a process, click the Add button. The Process Data dialog box opens:
	Name. Enter the exact name of the process, for example, java.exe.
	Port . Add a port number or name, either by typing a number or by clicking the Add button then selecting the ports in the Global Ports List.
	➤ If the process has to listen at a specific port, the port should be listed. You can enter more than one port, separated by commas, for example, 8888,8081,8080, 81,8000,82,80.
	➤ If the process does not have to listen at a specific port (that is, the software element can use any port), leave this field empty.
	Port match is optional. Select this check box to enable discovery of processes that are not listening on any of the ports entered in the Port field.
	Command Line. The software element can also be mapped using the process name. In this case, you must add a process command line (or part of it) with which the process name uniquely identifies the software, for example, c:\ora10\bin\oracle.exe UCMDB.
	Required . Select this check box if, during discovery, DDM must find all required processes (IP, port, command line, or owner), and each parse rule must be matched by at least one process.
	This feature is useful for applications that have shared processes running on their servers, and you need an identifying process that determines which type of server is being discovered. For example, HP Business Availability Center has shared processes running on the Gateway and Processing Servers and by selecting this check box, you can identify the correct server.
Software Element	The name of the definition.
Name	Note: This is not the software element's name but a name you give to differentiate this discovery from similar discoveries.

Software Library Dialog Box

Description	Enables you to view the logical groups of software elements.
	To access:
	➤ Run Discovery window > Network – Host Resources module > Software Element CF by Shell job. Locate the Global Configuration Files pane in the Properties tab. Select applicationsSignature.xml and click the Edit button.
	Manage Discovery Resources window >Host_Resources_Basic package >
	Dis_AppComponents_CF pattern. Locate the Global Configuration Files pane in the Pattern Signature tab. Select applicationsSignature.xml and click the Edit button. ➤ In the Infrastructure Wizard Preferences page, open the
	Choose Software Element to be discovered box.
Important Information	The software elements are organized in logical categories. You can change the names of these elements, you can move an element to another category, and you can define new elements and categories. For details, see the Category entry in "Software Identification Rule Editor Dialog Box" on page 245.
	The code you define in this dialog box and the Software Element Editor dialog box overwrites the code in applicationsSignature.xml.
Included in Tasks	"Discover Software Elements – Scenario" on page 200
Useful Links	"Global Configuration Files Pane" on page 237

GUI Element (A–Z)	Description
+ ✓ - ✓ -	Select a check box to include a category or software element in the discovery.
	Clear a check box to remove the category or element from the discovery.
+	Click to define a new software element. For details, see "Software Identification Rule Editor Dialog Box" on page 245.
×	Select a software element and click to delete the element.
⊘	Select a software element and click to make changes to the element. For details, see "Software Identification Rule Editor Dialog Box" on page 245.
<list elements="" of="" software=""></list>	List of objects that are software elements.

Chapter 6 • Manage Discovery Resources

Show Status Snapshot

This chapter provides information on viewing the current status of the discovered CIs in the DDM Probes.

This chapter includes:

Concepts

➤ Show Status Snapshot – Overview on page 251

Tasks

➤ View Current Status of Discovered CIs on page 252

Reference

➤ Show Status Snapshot User Interface on page 252

\lambda Show Status Snapshot – Overview

You use Show Status Snapshot to view the current status of the discovered CIs in the Probes. Show Status Snapshot retrieves the status from the Probes and displays the results in a view.



The view is not automatically updated; to refresh the status data, click the **Get snapshot** button.

View Current Status of Discovered Cls

This task describes how to view the current status of discovered CIs.

This task includes the following steps:

- ➤ "Prerequisites" on page 252
- ➤ "Access Show Status Snapshot" on page 252

1 Prerequisites

Verify that the Probe is enabled and is connected to the HP Business Availability Center Gateway server. For details, see "Install the DDM Probe" on page 36.

2 Access Show Status Snapshot

- **a** Go to **Discovery** > **Show Status Snapshot**.
- **b** Select a connected probe.
- **c** Click the Get Snapshot button.
- **d** Select jobs from the Progress list and click the **View Job progress** button. The Job Progress window opens.

🍳 Show Status Snapshot User Interface

This section describes:

- ➤ [Job Name] Dialog Box on page 253
- ➤ Show Status Snapshot Window on page 254

🙎 [Job Name] Dialog Box

Description	Enables you to view details about a job, including its scheduling, as well as job statistics.	
	To access: Select a job in the Progress pane of the Show Status Snapshot window and click the View job progress button.	
Important Information	Double-click a job to open a further dialog box with details of the job.	

GUI Element (A–Z)	Description		
Job Details	Status. Can be Scheduled (the job runs according to a defined schedule) or Running (the job is running now).		
	Last updated. The last time that the job was updated.		
	Threads. The number of threads currently allocated to this job.		
	Progress. The number of Trigger CIs in the job and the number of Trigger CIs that the Probe has finished working on.		
Schedule	Previous invocation. The last time that DDM ran the job		
	Next invocation. The next time that DDM is scheduled to run the job.		
	Last duration . The length of time, in seconds, taken to run the job in the previous invocation.		
	Average duration. The average duration, in seconds, of the time it took the Probe to run this job.		
	Recurrence. The number of times a job is run in a week. For example, if a job is scheduled to run daily, it runs 7 times in a week. If a job is scheduled to run weekly, Recurrence = 1.		
Statistics Results	For details, see "Statistics Results Pane" on page 256.		

💐 Show Status Snapshot Window

Description	Enables you to view the current status of discovered CIs and all active jobs running on the Probes. To access: Admin > Universal CMDB > Discovery > Show		
	Status Snapshot		
Important Information	Depending on what you select in the Domains and Probes pane, different information is displayed in the View pane.		
	If you select:		
	 a domain, you can view details and CIT statistics for the domain. For details, see "Details Pane" on page 169 and "Statistics Results Pane" on page 256. a Probe, you can view details on the Probe (such as the Probe III), the progress of a job and you can view CIT. 		
	Probe IP), the progress of a job and you can view CIT statistics. For details, see "Details Pane" on page 254, "Progress Pane" on page 255, "Statistics Results Pane" on page 256, and "View Pane" on page 257.		
Included in Tasks	"View Current Status of Discovered CIs" on page 252		
Useful Links	"Show Status Snapshot – Overview" on page 251		

Details Pane

GUI Element (A–Z)	Description	
Domain Type	Customer. A private domain used for your site. You can define several domains and each domain can include multiple Probes. Each Probe can include IP ranges but the customer domain itself has no range definition.	
	External. Internet/public domain. A domain which is defined with a range. The external domain may contain only one Probe whose name equals the domain name. However, you can define several external domains in your system. For details on defining domains, see "Add New Domain Dialog Box" on page 167.	

Progress Pane

GUI Element (A–Z)	Description	
0	Select a CI and click this icon to view details of a job. For details, see "[Job Name] Dialog Box" on page 253.	
Job	The name of the job. Double click a job to open a dialog box displaying job details. For details, see "[Job Name] Dialog Box" on page 253.	
Next invocation	The next time that the Probe is scheduled to run.	
Previous invocation	The last time that the Probe ran.	
Progress	Can be either Scheduled or Running. If a job is running, a progress bar displays the percentage finished.	
Thread count	The number of threads currently allocated to this job.	
Triggered CIs	The number of CIs triggered in the job.	

Statistics Results Pane

Description	Enables you to view details and CIT statistics.	
	To access: Click the Default Domain or Probe name in the Domains Browser pane.	

GUI Element (A–Z)	Description	
S	Click to retrieve the latest data from the Probe (data is not automatically updated).	
V	Set the time range for which to display statistics about the CITs.	
	➤ All. Displays statistics for all job runs.	
	➤ Last Hour/Day/Week/Month. Choose a period of time for which to display statistics about the CITs.	
	➤ Custom Range. Click to open the Customize Statistics Time Range dialog box. Enter the date or click the arrow to choose a date and time from the calendar, for the To and From dates. To delete a date, click Reset.	
<column title=""></column>	Click a column title to change the order of the CITs from ascending to descending order, or vice versa.	
<right-click a="" title=""></right-click>	Choose from the following options:	
	 Hide Column. Select to hide a specific column. Show All Columns. Displayed when a column is hidden. Customize. Select to display or hide columns and to change the order of the columns in the table. 	
	to change the order of the columns in the table. Opens the Columns dialog box.	
	➤ Auto-resize Column. Select to change a column width to fit the contents.	
	For details, see "Select Columns Dialog Box" in Reference Information.	

GUI Element (A–Z)	Description	
CIT	The name of the discovered CIT.	
Created	The number of CIT instances created by the Probe.	
Deleted	The number of CIT instances deleted by the Probe.	
Discovered CIs	The sum of all the CIs for all the invocations.	
Filter	The time range set with the Set Filter button.	
Last updated	The date and time that the statistics table has been updated for a particular Probe.	
Total	The total number of CIs in each column.	
Updated	The number of CIT instances that have been updated.	

View Pane

GUI Element (A–Z)	Description		
₹ 3	Click to view the current status of the discovered CIs and jobs on the selected Probe.		
Last updated	The date and time at which the Get snapshot button was last pressed (that is, the date and time of the data displayed in Status Snapshot).		
Probe IPs	The IP addresses defined for the Probe.		
Running jobs	The number of jobs running on the Probe.		
Scheduled jobs	The number of jobs that are scheduled to run according to the settings in the Discovery Scheduler. For details, see "Discovery Scheduler Dialog Box" on page 124.		
Status	The status of the Probe (either disconnected or connected).		
Threads	The sum of all threads currently allocated to the running jobs.		

Chapter 7 • Show Status Snapshot

Part III

Advanced Discovery

Discovery and Dependency Mapping Content

This chapter explains how to discover specific components on your system.

This chapter includes:

Concepts

- ➤ Discovery and Dependency Mapping Content Overview on page 262
- ➤ Class Model Overview on page 264

Tasks

- ➤ Application Microsoft Exchange on page 268
- ➤ Application SAP on page 276
- ➤ Application Siebel on page 282
- ➤ Application UDDI Registry on page 290
- ➤ Cluster Microsoft Cluster on page 292
- ➤ Cluster Veritas on page 294
- ➤ Database DB2 on page 297
- ➤ Database MS-SQL on page 299
- ➤ Database Oracle on page 301
- ➤ Discovery Tools on page 303
- ➤ Integration DDMi on page 303
- ➤ Integration NMM Layer 2 on page 303
- ➤ Integration StorageEssential on page 303

- ➤ J2EE WebLogic on page 304
- ➤ J2EE WebSphere on page 305
- ➤ Network Advanced on page 306
- ➤ Network Basic on page 308
- ➤ Network Credential-less on page 308
- ➤ Network Host Resources on page 315
- ➤ Network Layer 2 on page 320
- ➤ Network Load Balancer on page 335
- ➤ Network Connections Active Discovery on page 342
- ➤ Virtualization VMware on page 352
- ➤ Web Servers IIS on page 370

Discovery and Dependency Mapping Content – Overview

This chapter describes the discovery procedure for components in your systems, using the following sections:

- ➤ Overview. Provides a short explanation of what components are discovered.
- ➤ **Supported versions.** Lists the versions of the hardware or software that can be discovered.
- ➤ **Prerequisites.** Includes any setup or procedures that must be performed before running the discovery.
- ➤ **Network and protocols.** Lists the protocols that must be set up to enable discovery.
- ➤ **Discovered CITs.** Lists the CITs that are discovered.
- ➤ **Topology map.** Displays the topology map showing the components and their relationships.
- ➤ **Discovery workflow.** Provides the steps to be performed for discovery.
- ➤ **Troubleshooting and limitations.** Lists any issues that you should be aware of when running the discovery.

You can view information and descriptions for DDM elements as follows:

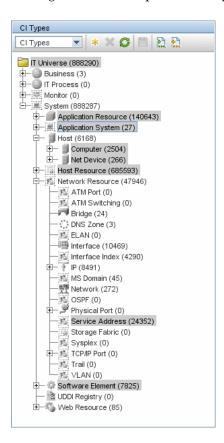
- ➤ CITs. Admin > Universal CMDB > Modeling > CI Type Manager. Select CI Types. For details, see "CI Types Overview" in *Model Management*.
- ➤ **Relationships.** Admin > Universal CMDB > Modeling > CI Type Manager. Select **Relationships**. For details, see "CI Type Relationships" in *Model Management*.
- ➤ Attributes. Admin > Universal CMDB > Modeling > CI Type Manager. Select the Attributes tab. Key attributes include an icon in the Key column. For details, see "CI Type Attributes" in *Model Management*.
- ➤ **Cardinality**. Admin > Universal CMDB > Modeling > Query Manager. Select an element and click the **Cardinality** tab in the Information pane.
- ➤ TQLs. Admin > Universal CMDB > Modeling > Query Manager. For details, see "Create a TQL Query" in *Model Management*.
- ➤ **Jobs.** Admin > Universal CMDB > Discovery > Run Discovery. For details, see "Run Discovery" on page 69.
- ➤ Patterns. Admin > Universal CMDB > Discovery > Manage Discovery Resources. For details, see Chapter 6, "Manage Discovery Resources."

Note:

- ➤ For details on modules that have been deleted from DDM, see "Discovery and Dependency Mapping API Changes" in the *HP Universal CMDB Deployment Guide* PDF.
- ➤ For details on viewing and analyzing Discovery results in Model Manager, see "Check Status of Application Discovery (Rediscover a View)" in *Model Management*.

Class Model – Overview

This section provides an overview of the UCMDB class model, focusing on the high-level and important CI Types.



This section includes the following topics:

- ➤ "The Host CIT" on page 265
- ➤ "The Host Resource CIT" on page 265
- ➤ "The Software Element CIT" on page 265
- ➤ "The Application Resource CIT" on page 266
- ➤ "The Application System CIT" on page 267
- ➤ "The Service Address CIT" on page 267

The Host CIT

The Host CIT represents any network node (physical or virtual) in your environment.

In UCMDB, there are two host CITs:

- ➤ **Computer.** A general purpose device, running general purpose operating systems (for example, UNIX, Windows, mainframe).
- ➤ **Network Device.** A network device with a pre-designated purpose (that is, Router, Switch, Load Balancer Device, Firewall, and so on).

Any host in UCMDB is identified either by an IP address or a strong ID value (such as a MAC address or hardware ID). Hosts identified by an IP address are considered incomplete (the Host Is Complete attribute has a false value). Hosts identified by a strong ID are considered complete.

The Host Resource CIT

The Host Resource CIT represents the resources of a host, which include both physical resources (for example, Memory, CPU) and OS resources (for example, File System, File, Network Share, OS User).

Host Resource CITs are always strongly contained in a Host CIT (they are linked with a container link), that is, they cannot exist outside the context of a host in the UCMDB world.

The Software Element CIT

The Software Element CIT represents a piece of software running on a host that is part of an application solution.

Examples of software elements discovered by DDM include: databases (Oracle, MSSQL, DB2, and Sybase), Web servers, Microsoft Exchange Server, J2EE servers, load balancing software, and cluster software.

Software Element CITs are considered either of a strong type or a weak type:

- ➤ Strong type Software Element CITs. These CITs are specialized (they are derived from Software Element) and include more attributes, different identification rules, a display label, and so on.
- ➤ Weak type Software Element CITs. These CITs are CI instances of the Software Element CI Type itself. They include basic configuration attributes only. Also, only one instance of each software component type can exist in the model. That is, you cannot model two different Oracle instances on a single host using a weak type software element.

A reconciliation mechanism handles the mapping of weak type software elements to their corresponding strong type, once they are discovered. This mechanism enables DDM to detect the existence of software by one method, and report additional details about the software when it is discovered using another method, knowing the two CIs will eventually be merged in the UCMDB.

The reconciliation mechanism relies on a shared attribute value between the weak type and strong type: the Name attribute (**data_name**). All software elements (strong or weak) are created with a distinguishing name: when a strong software element is discovered on a host containing a weak software element with the same name, the weak software element is merged with the strong one.

Weak type software elements are usually discovered by DDM's Host Resources discovery patterns. These patterns rely on a configuration file (Application Signature) that identifies key software according to the running processes and their network interconnections.

Like the Host Resource CIT, a Software Element CIT is strongly contained in a host.

The Application Resource CIT

The Application Resource CIT represents a software element's internal configuration. These CITs are used when attributes are not enough to describe the complex configuration of certain software elements. One group of such resources is the Database Resource group that includes CIs that describe a database's internal configuration (the database schema, database data files, database tables, and so on).

Further generic application resources (for example, Configuration Files, Resource Pool, Web Service) are included in multiple software element CITs, rather than being specific to a single CIT.

The Application Resource CIT is strongly contained in the Software Element CIT.

The Application System CIT

The Application System CIT represents an application deployment. Unlike the Software Element CIT, which is constrained to a one-host context, the Application System CIT is designed to represent application configurations across multiple hosts.

An Application System CI is usually linked via a member relationship to the Software Element CI. The Application CI together with the Software Element CI represent the application in the CMDB.

Examples of Application System CITs include Clusters, SAP System, Siebel Enterprise System, and MS Exchange System.

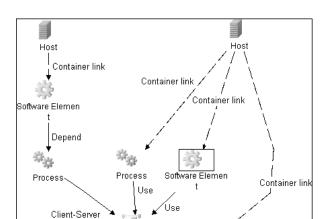
The Service Address CIT

The Service Address CIT derives from the Network Resource CIT and represents an address used by client software to retrieve a service from a software element.

The Service Address CIT key attributes include:

- ➤ Address Type. Currently acceptable values are TCP/UDP or URL.
- ➤ Service Address. The contents of the Service Address field depend on the address type. The TCP/UDP addresses use the format <IP-Address:Port>. The URL addresses contain the accessible URL value.

The mapping of software dependencies relies on the Service Address CIT in the model. The client Software Element is linked via a depend relationship to the Client Process CI. The Client Process CI has a Client Server relationship to a remote Service Address CIT. This Service Address CIT is connected through the use relationship to the process (Host Resource) and/or to the software element that provides a service at that address.



The Service Address CIT is also strongly contained in the Host CIT.

🦒 Application – Microsoft Exchange

This task explains how to discover Microsoft Exchange Servers.

This task includes the following steps:

Service Addre

- ➤ "Overview" on page 269
- ➤ "Supported Versions" on page 269
- ➤ "Prerequisites" on page 269
- ➤ "Network and Protocols" on page 269
- ➤ "Configuration Item Types" on page 269
- ➤ "Discovered CITs" on page 271
- ➤ "The Microsoft Exchange Server Package" on page 271
- ➤ "Views" on page 272
- ➤ "Topology Map" on page 272
- ➤ "Discovery Workflow" on page 273
- ➤ "Troubleshooting and Limitations" on page 274

1 Overview

DDM discovers the following components of Microsoft Exchange Server software: Microsoft Exchange System, Server, Administrative and Routing groups, Public folders and Folder trees, Links and Queues.

Currently, all information about the Microsoft Exchange server is retrieved by the WMI protocol from the **root\MicrosoftExchangeV2** namespace.

There are two jobs responsible for Microsoft Exchange Server discovery:

- ➤ Microsoft Exchange connection by WMI
- ➤ Microsoft Exchange topology by WMI

2 Supported Versions

Microsoft Exchange Server 2003 is supported.

3 Prerequisites

You must enable read-only permissions for the **root\MicrosoftExchangeV2 WMI** namespace. In some cases the **root\cimv2** namespace is also needed (with read-only permissions). For details, see "Troubleshooting and Limitations" on page 274.

4 Network and Protocols

Set up the **WMI protocol**. For details, see "WMI Protocol" on page 194.Information about Microsoft Exchange Server is taken from the **root\MicrosoftExchangeV2** namespace.

5 Configuration Item Types

The following CIs are created for Microsoft Exchange Server components:

Exchange

This CIT is located in the Application System folder. It is an abstract CIT that is the parent of the following CITs:

➤ Administrative group.

This CIT represents the administrative group in the Exchange system.

➤ Exchange System.

This CIT represents the top-level Exchange system. For example, if an organization uses the Exchange solution, then all the Exchange components are linked to a single Exchange system CI.

➤ Routing group.

This CIT represents routing groups existing in the Exchange system.

Microsoft Exchange Server

This CIT is located in the Software Element folder. The CIT represents Microsoft Exchange server software installed on a host.

Microsoft Exchange Resource

This CIT is located in the Application Resource folder. It is an abstract CIT that is the parent of the following CITs:

➤ Exchange folder.

This CIT represents the public folders available on the Exchange system. A public folder may be organized in an hierarchical structure, that is, one public folder may contain another public folder.

➤ Exchange folder tree.

This CIT provides information about public and private folder trees on Microsoft Exchange servers.

➤ Exchange link.

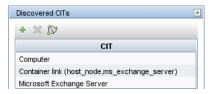
This CIT returns information about message-handling links between mail servers. A link can contain zero or more Exchange Queue objects, depending on the current message traffic along the link. In the Microsoft Exchange System Manager, these links are called queues.

➤ Exchange Message Queue.

This CIT represents queues available for Exchange links.

6 Discovered CITs

MS_Exchange_Connection_by_WMI



MS_Exchange_Topology_by_WMI



7 The Microsoft Exchange Server Package

All components responsible for Microsoft Exchange Server in DDM are bundled in the Microsoft_Exchange_Server package.

TQLs

Name	Category	Used by
ms_exchange_process_and _wmi	Trigger	Microsoft Exchange connection by WMI job
ms_exchange_server_and_ host_and_wmi	Trigger	Microsoft Exchange topology by WMI job
Microsoft Exchange topology	View	Microsoft Exchange topology view

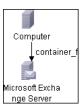
8 Views

The following view displays Microsoft Exchange Server components:

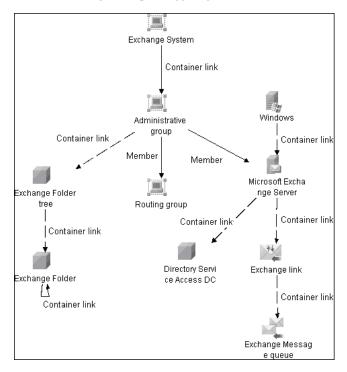
➤ Microsoft Exchange topology

9 Topology Map

MS_Exchange_Connection_by_WMI



MS_Exchange_Topology_by_WMI



10 Discovery Workflow

In the Run Discovery window, activate the following jobs:

- ➤ Network Basic (Host connection by WMI) to discover WMI CITs.
- ➤ Run any of the Network Host Resources jobs that gather information about processes running on a host. If a process named emsmta.exe is discovered on a host, the Microsoft Exchange connection by WMI job is triggered.
- ➤ Application Microsoft Exchange (Microsoft Exchange connection by WMI). The job reports the server that is actually running on this host. To discover other Exchange servers, you must run this job on each host where Microsoft Exchange is running. The job creates Microsoft Exchange Server CITs.

This job connects to the remote host by WMI to the **root\MicrosoftExchangeV2** namespace.

The following WMI queries are executed:

SELECT AdministrativeNote, CreationTime, ExchangeVersion, FQDN, GUID, MTADataPath, MessageTrackingEnabled, MessageTrackingLogFileLifetime, MessageTrackingLogFilePath, MonitoringEnabled, Type FROM Exchange_Server

This query returns all Exchange servers present on the Exchange system.

➤ Microsoft Exchange topology by WMI. The Exchange Server CI created by the Microsoft Exchange connection by WMI job acts as a trigger for this job. The Trigger CI connects to the host where the Exchange Server is running and retrieves the complete Microsoft Exchange Server topology. (For details on troubleshooting error messages, see "Troubleshooting and Limitations" on page 274.)

Chapter 8 • Discovery and Dependency Mapping Content

This job connects to the remote host by WMI to the **root\MicrosoftExchangeV2** namespace. The following WMI queries are executed (order is preserved):

SELECT AdministrativeGroup, DN, RoutingGroup FROM Exchange_Server SELECT AdministrativeGroup, AdministrativeNote, CreationTime, Description, GUID, Name, RootFolderURL FROM Exchange_FolderTree SELECT AddressBookName, AdministrativeNote, Comment, ContactCount, FolderTree, FriendlyUrl, IsMailEnabled, Path, Url FROM Exchange_PublicFolder SELECT Description, LDAPPort, Name, Type FROM Exchange_DSAccessDC SELECT GlobalStop, Linkld, LinkName, ProtocolName, Size, StateActive, StateFrozen, StateReady, StateRemote, StateRetry, StateScheduled, TypeCurrentlyUnreachable, TypeDeferredDelivery, TypeInternal, TypeLocalDelivery, VirtualMachine, VirtualServerName FROM Exchange_Link SELECT GlobalStop, Linkld, LinkName, ProtocolName, Queueld, QueueName, Size, VirtualMachine, VirtualServerName FROM Exchange Queue

11 Troubleshooting and Limitations

Administrative Group Limitation

If an Administrative group does not contain any Exchange servers or folder trees, the Administrative group is not discovered.

Error Messages

Error message	Reason	Solution
Failed to obtain host name	To model Exchange topology correctly the Microsoft Exchange connection by WMI job should know the name of the host to which it is connected. DDM tries to retrieve the host_hostname attribute of the host, matched by input TQL. If the attribute is not set, DDM runs the following WMI query to obtain the domain name of the host: SELECT Name FROM Win32_ComputerSystem If this query fails for any reason, the job also fails with this error message.	 Run any job that will retrieve the correct host name. Set the host name manually. Refer to the log files for more information as to why the WMI query for host name failed.
Failed to discover folder trees and public folders		Check if the credentials you use for connection match those described in "Prerequisites" on page 269.
Failed to discover domain controllers connection		Check if the credentials you use for connection match those described in "Prerequisites" on page 269.
Failed to discover links and queues	 Credentials are incorrect. Microsoft Exchange Server CIT used for connection: Does not have the FQDN property set. The host matched by the input TQL does not have the host_hostname property set. The WMI query for host name fails (see the Failed to obtain host name error message). 	 ➤ Make sure credentials are correct ➤ Make sure that ➤ The host_hostname property is set on the host running Microsoft Exchange ➤ The FQDN property is set on the Exchange Server ➤ The WMI query specified in the Failed to obtain host name error message returns the correct host domain name.

훹 Application – SAP

The SAP tasks discover either SAP ABAP or SAP Java. The Application Server ABAP provides the complete technology and infrastructure to run ABAP applications. The Application Server Java provides a Java 2 Enterprise Edition (Java EE) environment for developing and running Java EE programs.

Note: To discover more than one SAP system, it is recommended to create a SAP Protocol credential with a different user and password for each SAP system. For details on the SAP protocol and required user permissions, see "SAP Protocol" on page 182.

This section includes the following topics:

- ➤ "Discover SAP ABAP" on page 276
- ➤ "Discover SAP Java" on page 280



Discover SAP ABAP

This task discovers SAP ABAP architecture, SAP application components, SAP transactions and SAP Solution Manager business process definitions.

This task includes the following steps:

- ➤ "Overview" on page 276
- ➤ "Supported Versions" on page 277
- ➤ "Prerequisites" on page 277
- ➤ "Network and Protocols" on page 278
- ➤ "Discovery Workflow" on page 278

1 Overview

The SAP discovery process enables you to discover application components, SAP transactions and transports, and SAP topology.

2 Supported Versions

SAP BASIS and SAP AS (Architecture layer). Versions 3.x to 6.x.

SAP JCo. Version 3.0 (recommended). Note that DDM can discover SAP as long as the default SAP JCo provided with DDM is the correct version. If you are running an older version of SAP JCo, DDM may not be able to connect to SAP version 6.x.

SAP J2EE client. The version should match the relevant SAP system version.

3 Prerequisites

Install Java connectors:

➤ Download the SAP JCo package from the Tools & Services window of SAP JCo in SAP Service Marketplace:

https://websmp101.sap-ag.de/~form/sapnet? SHORTKEY=01100035870000463649

- ➤ Extract **sapjco-ntintel-2.0.8.zip** to a temporary directory (for example: C:\temp) on the Business Availability Center machine.
- ➤ Create a sap directory (in lowercase) in the C:\hp\DDM\DiscoveryProbe \root\ext\ directory on the machine where the Probe is installed.
- ➤ Copy sapjco.jar from the temporary directory to the C:\hp\DDM \DiscoveryProbe\root\ext\sap\ directory on the machine where the Probe is installed.
- ➤ Copy sapjcorfc.dll from the temporary directory to the %winnt%\system32 directory on the machine where the Probe is installed. Also copy the file to the C:\hp\DDM\DiscoveryProbe\root\ext\dll folder.
- ➤ Copy librfc32.dll from the temporary directory to the %winnt%\system32 directory. Also copy the file to the C:\hp\DDM\DiscoveryProbe\root\ext\dll folder.
- ➤ Verify that the MSVCR71.dll and MSVCP71.dll files are located in the %winnt%\system32 directory.

4 Network and Protocols

The following protocols enable connection to a machine to very whether a SAP system is installed on it:

- ➤ NTCMD Protocol. For details, see "NTCMD Protocol" on page 181.
- ➤ **SSH Protocol.** For details, see "SSH Protocol" on page 186.
- ➤ **Telnet Protocol**. For details, see "Telnet Protocol" on page 188.
- ➤ **SAP Protocol.** For details on required user permissions, see "SAP Protocol" on page 182.

5 Discovery Workflow

- **a** In the Run Discovery window, activate the modules in the following order:
 - ➤ Network Basic (Range IPs by ICMP, Host Connection By Shell).
 - ➤ Network Host Resources (Host Resources by Shell). This job discovers SAP software elements and processes.
 - ➤ Network Advanced (TCP Ports). Also, activate the SAP System by Shell job. This job discovers SAP J2EE Central Services and a SAP system without SAP J2EE credentials.
 - ➤ Web Servers Basic (Webserver Detection using TCP Ports). If the SAP system has an ITS configuration, to discover the ITS entities of the SAP system, run this job as a prerequisite to the SAP discovery that discovers ITS entities.

➤ Application – SAP

SAP System By Shell. This job searches for a SAP system by referring to the file system and process list. The SAP CI that is created is used as a trigger for the **SAP ABAP Connection by SAP JCO** job. This job needs Shell credentials and not SAP credentials.

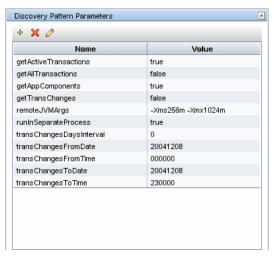
SAP ABAP Connection by SAP JCO. This job connects to the SAP system and creates a SAP System CI with a credentials ID. Subsequently, the other ABAP jobs use these credentials to connect to SAP.

SAP ABAP Topology by SAP JCO. Discovers infrastructure entities in the SAP system: hosts, application servers, work processes, databases, SAP clients, configuration files, software components (discovered as configuration files), and support packages (discovered as configuration files).

SAP Applications by SAP JCO. You run this job to discover the application components of this system. The result of this job may be many CIs. To omit unnecessary CIs, you can configure the pattern parameters as follows:

Access the SAP pattern: Manage Discovery Resources > SAP_Discovery package > Patterns > SAP_Dis_Applications.

Select the **Pattern Signature** tab and locate the **Discovery Pattern Parameters** pane.



Set one of the following parameters, and click **OK** to save the changes.

To discover all SAP transactions: Set **getAllTransactions** to **false**.

To discover active SAP transactions: Set **getActiveTransactions** to **true**.

To discover SAP transactions that have been changed by discovered transports:

Set **getTransChanges** to **true**;

set the from date (transChangesFromDate) and the to date (transChangesToDate). The date format is YYYYMMDD; set the from time (transChangesFromTime) and the to time (transChangesToTime). The time format is HHMMSS.

SAP ITS by NTCMD. Discovers Internet Transaction Server (ITS) entities (Application Gateway and Web Gateway).

SAP Solution Manager by SAP JCO. Discovers SAP Solution Manager components. SAP Solution Manager discovery enables you to discover the business process hierarchy.

- **b** For details on the CIs that are discovered, see the Statistics table in the Details tab, or click the **View Cls in Map** button. For details, see "Discovery Job Details Pane" on page 109.
- Verify that DDM discovered the appropriate components. Access the SAP ABAP TOPOLOGY view in View Manager and verify that the map displays all components.
- **d** To view the CIs discovered by the SAP discovery, access the Statistics Results pane, select a CI, and click the **View Instances** button, to open the **Discovered by** window. For details, see "Statistics Results Pane" on page 117 and "Discovered CIs Dialog Box" on page 119.

P Discover SAP Java

This task discovers the Application Server Java.

This task includes the following steps:

- ➤ "Overview" on page 281
- ➤ "Prerequisites" on page 281
- ➤ "Network and Protocols" on page 282
- ➤ "Discovery Workflow" on page 282

1 Overview

The SAP for Java discovery process enables you to discover SAP JAVA architecture and J2EE applications on the SAP JAVA server.

2 Prerequisites

- **a** Add the following JAR files to the C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeManager\discoveryResources\j2ee\sap directory on the Probe machine:
 - ➤ sapj2eeclient.jar
 - ➤ logging.jar
 - > exception.jar
 - ➤ sapxmltoolkit.jar

The files reside in the \usr\sap\<SID>\<instance name>\j2ee\j2eeclient directory on the SAP system machine.

b Add the com_sap_pj_jmx.jar file to the C:\hp\DDM\DiscoveryProbe\ root\lib\collectors\probeManager\discoveryResources\j2ee\sap directory on the Probe machine:

The file resides in the \usr\sap\<SID>\<instance name>\j2ee\admin\lib directory on the SAP system machine.

Note: If you create version folders under the \j2ee\sap directory on the Probe machine, you can connect to several SAP versions, by adding JAR files to each folder.

For example, to connect to versions 7.0 and 6.4:

- ➤ Create two folders under the **sap** folder.
- ➤ Name the folders **6.x** and **7.x**.
- ➤ Place the relevant jar files in these folders.

3 Network and Protocols

The following protocol enables connection to a machine and verification whether a SAP system is installed on it:

➤ **SAP JMX Protocol.** For details, see "SAP JMX Protocol" on page 181.

4 Discovery Workflow

- **a** In the Run Discovery window, activate the modules in the following order:
 - ➤ Network Basic (Range IPs by ICMP, Host Connection By Shell).
 - ➤ Network Host Resources (Host Resources by Shell). This job discovers SAP software elements and processes.
 - ➤ Network Advanced (TCP Ports). Also, activate the SAP System by Shell job. This job discovers SAP J2EE Central Services and a SAP system without SAP J2EE credentials.
 - ➤ Application SAP (SAP Java Topology by SAP JMX). This job discovers infrastructure entities in the SAP J2EE system: hosts, application servers, databases. Interfaces, Libraries, and Services are discovered as configuration files.

🦒 Application – Siebel

This task describes how to discover Siebel topology.

This task includes the following steps:

- ➤ "Overview" on page 283
- ➤ "Prerequisites Copy the driver Tool to the Probe Server" on page 284
- ➤ "Network and Protocols" on page 284
- ➤ "Discovered CITs" on page 285
- ➤ "Topology Map" on page 287
- ➤ "Discovery Workflow" on page 288

- ➤ "Copy the driver Tool and the SARM Analyzer Tool to the SiteScope Server" on page 289
- ➤ "Troubleshooting and Limitations" on page 290

1 Overview

Using the Siebel patterns, you can run an automatic Siebel discovery to create the Siebel world, together with its components, inside Business Availability Center.

During discovery:

- ➤ All Siebel-related IT entities that reside in the organization are discovered and configuration items (CIs) are written to the CMDB.
- ➤ When a new Siebel Application CI is created, two KPIs are created under it: Transactions and Locations.
- ➤ The relationships between the elements are created and saved in the CMDB.
- ➤ The newly generated CIs are displayed when the Siebel Enterprises view is selected in View Explorer under the Siebel Enterprises root CI.
- ➤ Four logical containers—Applications, Business Processes, Hosts and Locations—are created under the Siebel Enterprises root CI.
- ➤ After discovery has run, you must manually update some of the discovered CI's properties. For details, see "Configure HP Business Availability Center for Siebel Applications" in *Solutions and Integrations*.

Note: Verify that all Siebel server IP addresses are included in the range. If not all servers can be covered with one IP range, you can split the range into several ranges.

2 Prerequisites - Copy the driver Tool to the Probe Server

The driver tool is used to extract data about the enterprise structure from Siebel.

Note: If you are working with different versions of Siebel in your organization, make sure you use a driver tool with a version that is appropriate for the Siebel server.

To copy the driver tool to the Probe server:

- **a** Copy the driver Command Line Interface (CLI) tool from the Siebel server to any folder on the Probe server.
- **b** It is recommended to run the Siebel connection test to validate the driver installation. To run the connection test, open the command line on the Probe server and change directory to the location of the driver.exe file.
- **c** Run from the command line:

>driver /e [site_name] /g [gateway_host] /u [username] /p [password]

If the connection is established successfully, the Command Prompt window displays the driver prompt and a status message about the number of connected servers.

3 Network and Protocols

Set up the following protocols for the Windows platform and continue to Discovery Workflow:

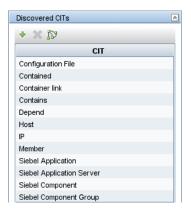
- ➤ WMI Protocol. For details, see "WMI Protocol" on page 194.
- ➤ NTCMD Protocol. For details, see "NTCMD Protocol" on page 181.
- ➤ **Siebel Gateway Protocol.** For details, see "Siebel Gateway Protocol" on page 183.

Set up the following protocols for the UNIX platform and continue to Discovery Workflow:

- ➤ **SSH Protocol.** For details, see "SSH Protocol" on page 186.
- ➤ **Telnet Protocol.** For details, see "Telnet Protocol" on page 188.
- ➤ **Siebel Gateway Protocol.** For details, see "Siebel Gateway Protocol" on page 183.

4 Discovered CITs

SIEBEL_DIS_APP_SERVERS



SIEBEL_DIS_APP_SERVER_CONFIG



SIEBEL_DIS_GATEWAY_CONNECTION_(GTWY)



SIEBEL_DIS_WEBAPPS_UNIX



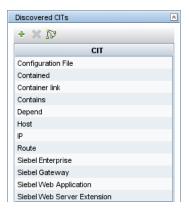
SIEBEL_DIS_DB_UNIX



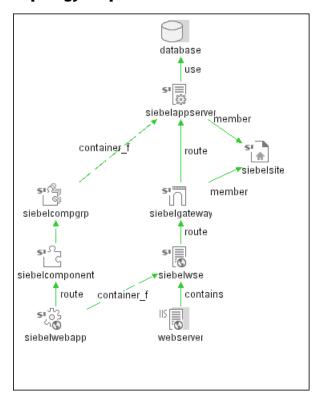
SIEBEL_DIS_DB_NT



SIEBEL_DIS_WEBAPPS_NT



5 Topology Map



6 Discovery Workflow

- **a** For Siebel discovery to run, you must copy the driver tool to the Probe server. For details, see "Prerequisites Copy the driver Tool to the Probe Server" on page 284.
- **b** For the SiteScope Siebel monitors to work correctly, copy the driver tool and the SARM Analyzer tool to the SiteScope server. On SiteScope, the driver tool is launched by the Siebel Application Server monitor to retrieve the metrics. For details, see "Copy the driver Tool and the SARM Analyzer Tool to the SiteScope Server" on page 289.
- **c** To trigger the discovery of Siebel networking features, add a Network CI to the CMDB. For details, see "New CI Dialog Box" in *Model Management*.
- **d** In the Run Discovery window, activate the modules in the following order:
 - ➤ Network Basic (Class C IPs by ICMP, Host Connection by WMI)
 - ➤ Application Siebel (Siebel DB by TTY)
- **e** To discover the Web tier, activate the following modules:
 - ➤ Network Advanced (TCP Ports)
 - ➤ Application Siebel (Siebel Web Applications by NTCMD, Siebel Web Applications by TTY, Siebel DB by WMI and NTCMD)
 - ➤ Web Server Basic (WebServer Detection using TCP Ports)
- **f** To discover Siebel, activate all the patterns in the Application Siebel module.

Note: The following enrichment patterns automatically run in the background during discovery:

Siebel_Route_WebApp_To_Component. Builds the route between Siebel Web Application CIs and Siebel Component CIs.

Siebel_Web_To_Middle_Tier. Builds the route between the Web tier and the middle tier when the Siebel enterprise uses a Resonate server for load balancing.

g For details on the CIs that are discovered, see the Statistics table in the Details tab.

7 Copy the driver Tool and the SARM Analyzer Tool to the SiteScope Server

Note:

- ➤ The SARM Analyzer is used for analyzing SARM data, so that it can be displayed in the Business Availability Center for Siebel SARM User Trace Breakdown tab.
- ➤ If you are working with different versions of Siebel in your organization, make sure you use a driver and a SARM Analyzer with a version that is appropriate for the SiteScope server.
- ➤ It is recommended to run the Siebel connection test to validate the driver installation.
- ➤ SiteScope should run under a domain user name which has permissions to run server manager and SARM Analyzer and also has read access to the log folders on the Siebel servers (Web servers and application servers).
- ➤ All Siebel servers (Windows and UNIX) must be defined as remote servers on the SiteScope server. For details, refer to the SiteScope Reference Guide.

To copy the driver tool to the SiteScope server:

- **a** Copy the driver Command Line Interface (CLI) tool from the Siebel server to any folder on the SiteScope server.
- **b** To run a connection test, open the command line on the SiteScope server and change directory to the location of the driver.exe file.
- **c** Run from the command line:

>driver /e [site_name] /g [gateway_host] /u [username] /p [password]

For the connection to work properly, verify that the user and password have the correct permissions for a remote connection.

If the connection is established successfully, you should see the driver prompt and the status message about the number of connected servers.

To copy the SARM Analyzer tool to the SiteScope server:

- **a** Copy the SARM Analyzer tool from the Siebel server to a folder on the SiteScope server.
- **b** If your site includes a large number of Web servers, it is preferable to use multiple SiteScopes to distribute the work done by the SARM Analyzer tool between those SiteScopes. In such a case, copy SARM Analyzer to each SiteScope.

8 Troubleshooting and Limitations

The Siebel DB by TTY job cannot discover virtual Siebel application servers (with a different name and configuration to the actual Siebel application server) running on UNIX machines.

🖒 Application – UDDI Registry

This task describes how to discover UDDI processes.

This task includes the following steps:

- ➤ "Overview" on page 290
- ➤ "Network and Protocols" on page 291
- ➤ "Topology Map" on page 291
- ➤ "Discovery Workflow" on page 291

1 Overview

The UDDI discovery process enables you to discover Web services from a UDDI registry.

DDM queries the UDDI registry for its Web services, including non-SOAP services, or for a specific publisher service (if defined in the UDDI Registry protocol). The Web services found in the UDDI registry are represented by a **webservice** CI in the CMDB and the registry is created as a **uddiregistry** CI.

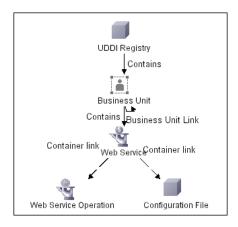
Note: Business Availability Center supports UDDI versions 2 and 3.

2 Network and Protocols

- **a** Set up the **UDDI protocol**. For details, see "UDDI Registry Protocol" on page 190.
- **b** In the Run Discovery window, activate the **Application Webservices** module.
- For details on the CIs that are discovered, see the Statistics table in the Details tab.

3 Topology Map

The following depicts the topology of the **SOA UDDI View**:



4 Discovery Workflow

(Optional) To enter the name of the service publisher whose services must be published:

- a Access the Manage Discovery Resources window.
- **b** In the Discovery Resources pane, locate the Webservices package and select the **UDDI_Registry** pattern.

- **c** In the Pattern Signature tab, in the Discovery Pattern Parameters pane, select the **organization** parameter and click the **Edit** button.
- **d** In the Parameter Editor:
 - ➤ In the **Value** box, enter the name of the service publisher.
 - ➤ In the **Description** box, enter the required description of the organization.
- **e** Save the changes.

🏲 Cluster – Microsoft Cluster

This task describes how to discover Microsoft Cluster servers.

This task includes the following steps:

- ➤ "Overview" on page 292
- ➤ "Network and Protocols" on page 292
- ➤ "Topology Map" on page 293
- ➤ "Discovery Workflow" on page 293

1 Overview

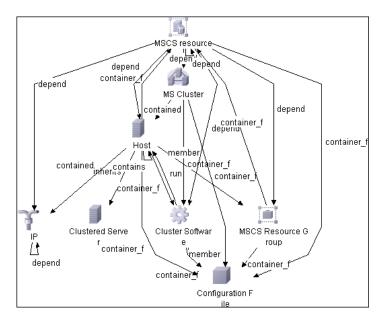
The MS Cluster discovery process enables you to discover the topology of a Microsoft Cluster Server on the network.

2 Network and Protocols

- ➤ WMI Protocol. For details, see "WMI Protocol" on page 194.
- ➤ NTCMD Protocol. For details, see "NTCMD Protocol" on page 181.

3 Topology Map

The Microsoft Cluster Server View shows the MS Cluster and the cluster software (the agents running on the actual host) as its members. The cluster is composed of several Clustered Servers that are the virtual hosts or servers providing the platform for the virtual service used by the cluster clients (through the virtual IPs). The cluster contains Microsoft Cluster Groups. Each of the groups contains Microsoft Cluster Resources. For each Cluster Resource Group, we assume that different, dedicated virtual IPs are being assigned; these IPs are configured for the use of the cluster clients.



4 Discovery Workflow

- **a** In the Run Discovery window, activate the modules in the following order:
 - ➤ Network Basic (Host Connection by WMI)
 - ➤ Network Host Resources
 - ➤ Cluster Microsoft Cluster
- **b** For details on the CIs that are discovered, see the Statistics table in the Details tab.

Teluster – Veritas

This task describes how to discover Veritas Cluster servers.

This task includes the following steps:

- ➤ "Overview" on page 294
- ➤ "Network and Protocols" on page 294
- ➤ "Topology Map" on page 294
- ➤ "Discovery Workflow" on page 296

1 Overview

The Veritas Cluster discovery process enables you to discover Veritas Cluster Servers (VCS), and their member machines (also referred to as nodes), that activate the discovered resources provided by the cluster.

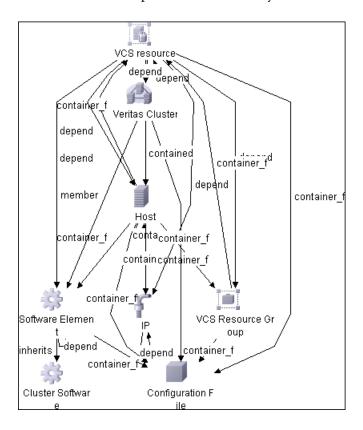
2 Network and Protocols

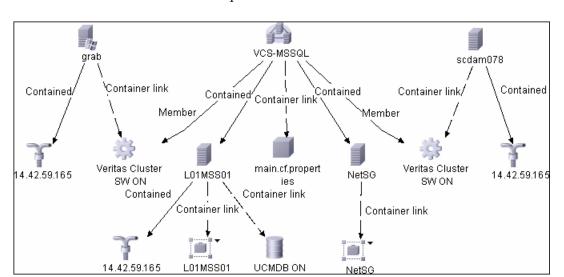
- **a** Set up the **SSH protocol**. For details, see "SSH Protocol" on page 186.
- **b** Set up the **Telnet protocol**. For details, see "Telnet Protocol" on page 188.

3 Topology Map

This view shows the top layer of the Veritas Cluster topology. It displays the discovered Veritas Cluster and the clustered software resources that are members of that cluster. Each software resource is linked by a **member** relationship to the Veritas Cluster.

Veritas Clusters are composed of several clustered servers. Each server is responsible for running certain services and applications. The servers are used as backups for one another. When a system components fails, another server takes over to provide the necessary service.





Double-click the required node to drill down to the CIs folded underneath.

A Veritas Cluster group is a collection of dependent or related resources that is managed as a single unit. Each Veritas Cluster group is linked to a designated node, which is responsible for activating the resources contained in the group. If a failure occurs in the designated node, the responsibility for activating the resources is switched over to a different node.

4 Discovery Workflow

In the Run Discovery window, activate the following modules:

- ➤ Network Protocol Connections
- ➤ Host Resources SSH/Telnet
- ➤ Veritas Cluster

For details on the CIs that are discovered, see the Statistics Results table in the Details tab. For details, see "Statistics Results Pane" on page 117.

🦒 Database – DB2

This task describes how to discover IBM DB2 databases.

This task includes the following steps:

- ➤ "Overview" on page 297
- ➤ "Topology Map" on page 297
- ➤ "Network and Protocols" on page 297
- ➤ "Discovery Workflow" on page 298
- ➤ "Troubleshooting and Limitations" on page 298

1 Overview

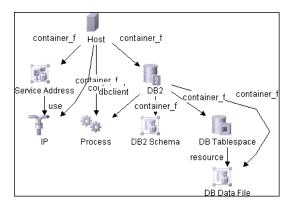
Discovers IBM DB2 Server databases and their components on the network.

2 Network and Protocols

IBM DB2 Server uses the **SQL protocol**.

3 Topology Map

The following image depicts the topology of the IBM DB2 Server view:



This view shows a host on which an IBM DB2 Server and DB2 Schema are installed, the processes that communicate with the server (connected by DB Client links), and the DB tablespaces.

4 Discovery Workflow

- **a** Verify the user name, password, and port used by IBM DB2 Server.
- **b** Set up the **SQL protocol**. For details, see "SQL Protocol" on page 185. In the Database Type box, choose **db2**.
- **c** In the Run Discovery window, activate the jobs in the **Database DB2** module in the following order:
 - ➤ DB2 Connection by SQL
 - ➤ DB2 Topology by SQL
 - ➤ Databases TCP Ports
- **d** For details on the CIs that are discovered, see the Statistics table in the Details tab. For details, see "Statistics Results Pane" on page 117.

5 Troubleshooting and Limitations

To perform an IBM DB2 discovery, copy the following files from the installation folder on the IBM DB2 machine to the DDM Probe machine:

- ➤ db2java.zip
- ➤ db2jcc.jar
- ➤ db2jcc_license_cisuz.jar
- ➤ db2jcc_license.jar

Place the files in the following folder: C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeManager\discoveryResources\db\db2.

Restart the Probe.

🚏 Database – MS-SQL

This task describes how to discover the Microsoft SQL Server application.

This task includes the following steps:

- ➤ "Overview" on page 299
- ➤ "Supported Versions" on page 299
- ➤ "Topology Map" on page 300
- ➤ "Network and Protocols" on page 299
- ➤ "Discovery Workflow" on page 300

1 Overview

You can discover Microsoft SQL Server databases and their components on your network.

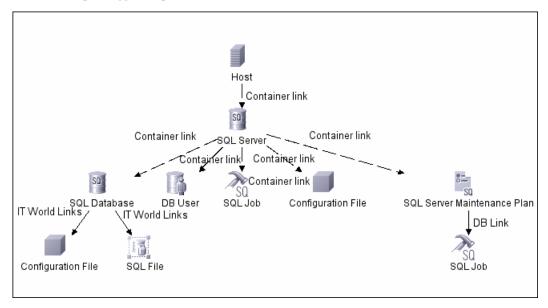
2 Supported Versions

Microsoft SQL Server 2000, 2005.

3 Network and Protocols

Microsoft SQL Server uses the **SQL protocol**. For details, see "SQL Protocol" on page 185.

4 Topology Map



This view shows the hosts on which Microsoft SQL Server is installed. Microsoft SQL Server contains the databases, users, SQL jobs, and configuration files of this database, and maintenance plans.

5 Discovery Workflow

- **a** Verify the user name, password, and port used by Microsoft SQL Server.
- **b** Set up the **SQL protocol**. For details, see "SQL Protocol" on page 185.
- **c** In the Run Discovery window, activate the jobs in the **Database** − **MS-SQL** module in the following order:
 - ➤ Databases TCP Ports
 - ➤ MSSQL Connection by SQL
 - ➤ MSSQL Topology by SQL
- **d** For details on the CIs that are discovered, see the Statistics table in the Details tab. For details, see "Statistics Results Pane" on page 117.

Database – Oracle

This task describes how to discover Oracle databases.

This task includes the following steps:

- ➤ "Overview" on page 301
- ➤ "Supported Versions" on page 301
- ➤ "Prerequisites" on page 301
- ➤ "Network and Protocols" on page 301
- ➤ "Discovered CITs" on page 301
- ➤ "Topology Map" on page 302

1 Overview

This discovery adds a valid credentials ID to the CMDB. You can then use this CI to fully discover the database.

2 Supported Versions

Oracle 8, 9, 10.

3 Prerequisites

Run **Databases TCP Ports**. (Open TCP\UDP ports on a host with known server ports are discovered.)

4 Network and Protocols

To discover Oracle databases, use the following protocol:

➤ **SQL.** For details, see "SQL Protocol" on page 185.

5 Discovered CITs

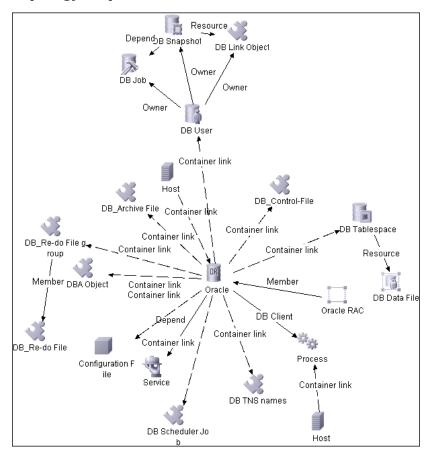
➤ owner, dbjob, dbuser, process, dbclient, dblinkobj, dbsnapshot, dbdatafile, dbtablespace, db_controlfile, db_redofile, db_redofilegroup, db_archivefile, oracle, dbschedulerjob, service, rac

Chapter 8 • Discovery and Dependency Mapping Content

The following attributes are updated:

- ➤ database_dbversion
- ➤ database_dbtype
- ➤ database_dbsid
- ➤ database_dbport

6 Topology Map



Discovery Tools

This module holds the jobs necessary to:

- ➤ Discover document files and directories.
- ➤ Discover hosts using the Nslookup command on the Shell of every DNS server in the scope.
- ➤ Serve as an example of dynamically creating and using credentials for connecting to remote machines.
- ➤ Import data from external sources, for example, CSV files, properties files, and databases. For details, see Chapter 9, "Importing Data from External Sources."

훩 Integration – DDMi

For details on the procedure for integration between HP Universal CMDB and DDMi, see the *HP Universal CMDB–Data Dependency and Mapping Inventory (DDMi) Integration Guide* PDF.

🚏 Integration – NMM Layer 2

For details on the procedure for integration between HP Universal CMDB and NNMi, see the *HP Universal CMDB-HP Network Node Manager i (NNMi)*Integration Guide PDF.

This is a second of the second

For details on the procedure for integration between HP Universal CMDB and Storage Essentials, see the HP Universal CMDB—Storage Essentials (SE) Integration Guide PDF.

🖒 J2EE – WebLogic

This task describes how to discover WebLogic applications.

This task includes the following steps:

- ➤ "Overview" on page 304
- ➤ "Supported Versions" on page 304
- ➤ "Prerequisites" on page 304
- ➤ "Network and Protocols" on page 305
- ➤ "Discovery Workflow" on page 305

1 Overview

DDM first finds WebLogic servers based on the JMX protocol, then discovers the WebLogic J2EE environment and components.

The WebLogic discovery process enables you to discover all the deployed Web services and operations deployed on a WebLogic server.

2 Supported Versions

The following versions are supported: WebLogic 6.0, 6.1, 7.0, 8.1, 9.0, 9.1, 9.2, and 10.

3 Prerequisites

If you are using WebLogic version 7, perform the following procedure to discover this Weblogic version:

- ➤ Take the webserviceclient.jar and weblogic.jar files from the following location: <BEA Installation root folder>\<WebLogic version number>\server\lib.
- ➤ Place both JAR files in the following location: C:\hp\DDM\DiscoveryProbe \root\lib\collectors\probeManager\discoveryResources. (This folder is created when the Probe connects to the WebLogic server. However, if you have not yet run DDM, you must manually create the folder.)
- ➤ Rename the JAR files by adding a suffix that includes the WebLogic version number, as follows: webserviceclient70.jar, weblogic70.jar.

4 Network and Protocols

Set up the **WebLogic protocol**. For details, see "WebLogic Protocol" on page 191.

5 Discovery Workflow

- ➤ In the Run Discovery window, activate the job in the J2EE WebLogic module.
- ➤ For details on the CIs that are discovered, see the Statistics Results table in the Details tab. For details, see "Statistics Results Pane" on page 117.

р J2EE – WebSphere

This task describes how to discover WebSphere applications.

This task includes the following steps:

- ➤ "Overview" on page 305
- ➤ "Supported Versions" on page 305
- ➤ "Network and Protocols" on page 306
- ➤ "Discovery Workflow" on page 306

1 Overview

DDM first finds WebSphere servers based on either SOAP or RMI authentication, then discovers the WebSphere J2EE environment and components.

WebSphere discovery discovers Web services that are deployed on an IBM WebSphere server. The discovered Web services are represented by the webservice CIT in the CMDB.

2 Supported Versions

Business Availability Center supports WebSphere versions 5 and 6.

3 Network and Protocols

Set up the **WebSphere protocol**. For details, see "WebSphere Protocol" on page 193.

4 Discovery Workflow

- **a** In the Run Discovery window, activate the **J2EE WebSphere** job.
- **b** For details on the CIs that are discovered, see the Statistics Results table in the Details tab. For details, see "Statistics Results Pane" on page 117.

Network – Advanced

This task describes how to discover DNS zones.

This task includes the following steps:

- ➤ "Overview" on page 306
- ➤ "Network and Protocols" on page 306
- ➤ "Discovered CITs" on page 307
- ➤ "Topology Map" on page 307
- ➤ "Discovery Workflow" on page 307
- ➤ "Troubleshooting and Limitations" on page 307

1 Overview

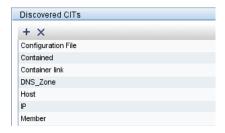
You can discover the Domain Name System (DNS) server topology, that is, the zones that each server manages, and the IPs in each zone.

- ➤ The pattern is triggered on every DNS with WMI: DDM activates the registry query to retrieve all the zones that this DNS manages.
- ➤ DDM queries each zone by activating Nslookup on the Probe against the DNS servers.

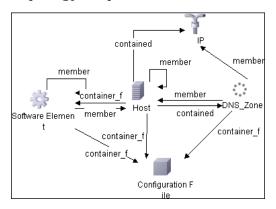
2 Network and Protocols

Nslookup (Zone transfer)

3 Discovered CITs



4 Topology Map



5 Discovery Workflow

In the Run Discovery window, activate the jobs in the following order:

- ➤ DNS Zone by Nslookup
- ➤ NTCMD connection pattern on the environment to discover the DNS servers
- ➤ Connect to these machines by WMI
- ➤ Run the pattern on the WMI that was discovered on these machines

6 Troubleshooting and Limitations

Check that Nslookup is permitted in the environment. In many environments, Nslookup is blocked by the network administrator, in which case DDM cannot retrieve data.

Network - Basic

You activate the jobs in the network modules to discover information about host components, for example, which hosts have open TCP/UDP ports, the ARP table of a router that is using the SNMP protocol, and so on.

For details on using a wizard to discover the network, see "Infrastructure Wizard" on page 128.

🦒 Network – Credential-less

This task describes how to use the **Host Fingerprint using nmap** job to discover hosts, operating systems, network interfaces, applications, and running services.

This task includes the following steps:

- ➤ "Overview" on page 308
- ➤ "Prerequisites" on page 308
- ➤ "Pattern Parameters" on page 313
- ➤ "Discovered CITs" on page 314
- ➤ "Discovery Workflow" on page 314
- ➤ "Troubleshooting and Limitations" on page 314

1 Overview

Nmap is a utility for network exploration that uses raw IP packets to determine which hosts are available on the network, which services those hosts are offering, which operating systems they are running on, and so on.

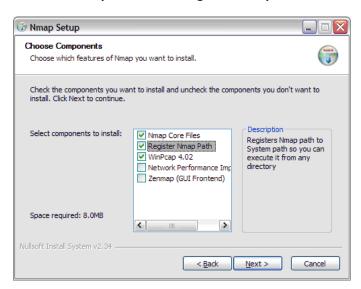
Nmap also calculates to what extent the operating system result is accurate, for example, 80% accuracy. The Host Fingerprint using nmap job, which relies on the Nmap utility, reports the Nmap accuracy value on the host_osaccuracy attribute on the Host CI.

2 Prerequisites

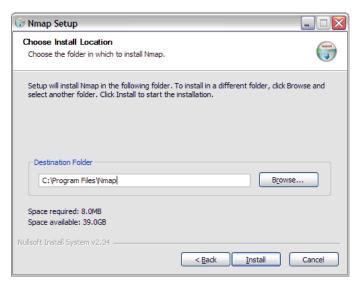
Use the following procedure to install Nmap and WinPcap.

Perform the following procedure on every Probe machine that will run the Host Fingerprint using nmap job:

- a Run nmap-4.76-setup.exe fromC:\hp\DDM\DiscoveryProbe\nmap_install.
- **b** Accept the terms of the license and click **I agree**. The **Choose Components** dialog box opens.
- c Select Nmap Core Files, Register Nmap Path, and WinPcap 4.02.





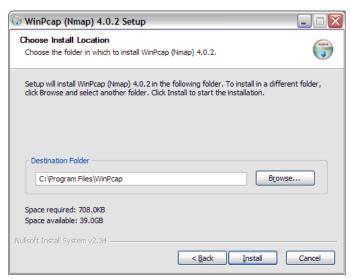


a Accept the default location or enter another location.

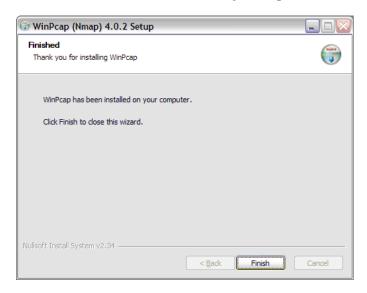
Click Install.

Nmap is installed. The WinPcap installation dialog box opens immediately after the Nmap installation is complete.

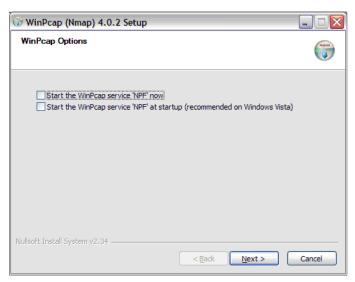
b Accept the terms of the license and click **Next**. The **Choose Install Location** dialog box opens.



c Accept the default location or enter another location. Click **Install**. The Finished dialog box opens.



Click Finish. The WinPcap Options dialog box opens.



- **d** Clear the check boxes and click **Next**.
- e Click Finish.

The following software is added to the Probe machine:

- ➤ Nmap 4.76
- ➤ winpcap-nmap 4.02
- ➤ Microsoft Visual C++ Redistributable x86 9.0.21022

To verify, access the Add/Remove Programs window.

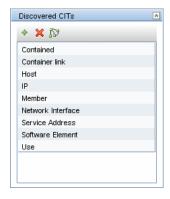
3 Pattern Parameters

To view the pattern parameters: Run Discovery > Network – Credentialless Discovery > Host Fingerprint using nmap > Properties tab > Parameters pane. For details on overriding parameters, see "Parameters Pane" on page 149.

Override	Name	Value
	Create_Application_CI	1
	Perform_Port_Fingerprints	1
	nmap_host_timout	600
	scan_known_ports_only	0
	scan_these_ports_only	

- ➤ Create_Application_CI. Creates an application CI based on the port fingerprint information.
- ➤ Perform_Port_Fingerprints. Tries to discover opened ports.
- ➤ nmap_host_timeout. The length of time Nmap is allowed to spend scanning a single host (in seconds).
- ➤ scan_known_ports_only. Scans for ports listed in the portNumberToPortName.xml file.
- ➤ scan_these_ports_only. Limits the range of ports to be scanned, for example, T:1-10,42,U:1-30 (discover TCP ports 1 to 10 and 42 and UDP ports 1-30). If this parameter is left empty, the Nmap default is used.
- ➤ discover_os_name. Discovers host OS, which may have some inaccuracy.

4 Discovered CITs



5 Discovery Workflow

This job is triggered on any discovered IP address.

6 Troubleshooting and Limitations

Error Message	Reason	Solution
Can't parse XML document with Nmap results. Skipped.	nmap.exe failed before it could create a valid XML file.	 Try to restart the Nmap job. Try to reduce the number of threads for the Nmap job.
Error nmap result file is missing	nmap.exe failed before it could create an XML file.	 Try to restart the Nmap job. Try to reduce the number of threads for the Nmap job.
The system cannot execute the specified program (in the communication log file)	The Windows system cannot launch the Nmap application.	Verify that: ➤ The correct Nmap version has been downloaded and installed. ➤ WinPcap has been installed. For details on these installations, see "Prerequisites" on page 308. If you have installed Nmap and WinPcap, and the error message still appears in the communication log, install vcredist_x86.exe from C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeManager\discoveryResources.

Error Message	Reason	Solution
Nmap is not installed on Probe machine	Nmap is not installed on the Probe machine.	Try to launch Nmap from the command line. Make sure that Nmap is installed. For details on the installation, see "Prerequisites" on page 308.

🏲 Network – Host Resources

This task describes how to discover host resources.

This task includes the following steps:

- ➤ "Overview" on page 315
- ➤ "Prerequisites" on page 315
- ➤ "Network and Protocols" on page 316
- ➤ "Discovered CITs" on page 317
- ➤ "Topology Map" on page 319
- ➤ "Discovery Workflow for the Software Element CF by Shell Job" on page 319
- ➤ "Troubleshooting and Limitations" on page 319

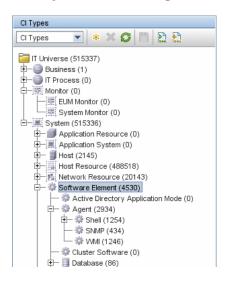
1 Overview

You can discover resources on Windows and UNIX hosts, for example, disk information, running processes or services, and so on.

2 Prerequisites

- **a** As DDM takes information from the Probe database, before running the Software Element CF by Shell job, you must first run the following jobs:
 - ➤ Host Resource. To discover the processes, activate Host Resources by SNMP, Host Resources by Shell, or Host Resources by WMI.
 - ➤ Network Connections Active Discovery. To discover open ports, process to port relationships and TCP connections, activate **Collect Network Data By Shell or SNMP.** For details, see "Network Connections – Active Discovery" on page 342.

- **b** Verify that the **Oracle Credentials by SQL** job is activated. (Note that this job uses the **oracle_database_no_user** Trigger TQL.)
- c Verify that the CMDB already contains the Agent and Shell CITs: Modeling > CI Type Manager. Search for Software Element, and verify that Agent and Shell are present:



3 Network and Protocols

To discover host resources, define the following protocols:

- ➤ NTCmd. For details, see "NTCMD Protocol" on page 181.
- ➤ **SNMP**. For details, see "SNMP Protocol" on page 183.
- ➤ **SSH/Telnet.** For details, see "SSH Protocol" on page 186 and "Telnet Protocol" on page 188.
- ➤ WMI. For details, see "NTCMD Protocol" on page 181.

Users do not need root permissions, but do need the appropriate credentials to enable connecting to the remote machines and running the relevant commands.

4 Discovered CITs

➤ The following CITs are discovered by the Host Resources by Shell job:



➤ The following CITs are discovered by the Host Resources by SNMP job:



➤ The following CITs are discovered by the Host Resources by WMI job:

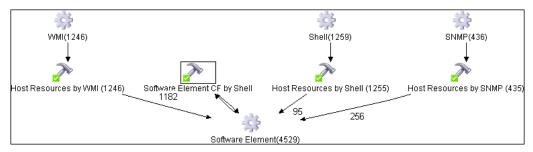


➤ The following CITs are discovered by the Software Element CF by Shell job:



The attributes for the resource name and its root container (that is, its parent) are updated during the run.

5 Topology Map



6 Discovery Workflow for the Software Element CF by Shell Job

In the Run Discovery window, activate the job (**Discovery Modules > Network – Host Resources > Software Element CF by Shell**). This job retrieves the software element's configuration file and maps the file to the correct application by referring to the applicationsSignature.xml file. The triggered CIs are software elements that have Shell running on their host and that include a configuration file definition that matches the definition in the applicationsSignature.xml file.

For details on discovering Oracle configuration files, see "Discover Software Elements – Scenario" on page 200.

7 Troubleshooting and Limitations

Problem. The error message **No data processed** is displayed.

Solution. Either run or rerun Host Resources by Shell/SNMP/WMI and Collect Network Data by Shell or SNMP.

🦒 Network – Layer 2

Note: Layer 2 discovery runs on Catalyst (Cisco Systems) network switches only.

This task describes how to discover Layer 2 objects.

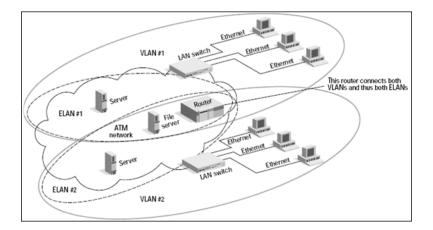
This task includes the following steps:

- ➤ "Overview" on page 320
- ➤ "Prerequisites" on page 321
- ➤ "Discovered CITs" on page 322
- ➤ "Layer 2 Relationships" on page 323
- ➤ "Topology Map" on page 324
- ➤ "Discovery Workflow" on page 325
- ➤ "Troubleshooting and Limitations" on page 334

1 Overview

The Layer 2 package discovers the Layer 2 topology that includes the switches tree topology (the backbone links between the switches) and also the end user connections to the switch-ports (the Layer 2 links between a switch and a host). The Layer 2 package is based on the SNMP protocol.

The following image illustrates a router connecting overlapping VLANs/ELANs:



2 Prerequisites

Important:

- ➤ All network connection patterns should finish running before you activate the Layer 2 patterns.
- ➤ Make sure that there is SNMP access to all switches in the environment to be discovered, as that is a key requirement for fully discovering the Layer 2 topology.
- ➤ When defining the SNMP protocol credentials, have available the Port and Community authentication parameters.
- ➤ In the Network Layer 2 module, run the Host Networking By SNMP job. As a result of this run, DDM saves SNMP CIs to the CMDB.

Note: Layer 2 discovery is based on the connection patterns for the following reasons:

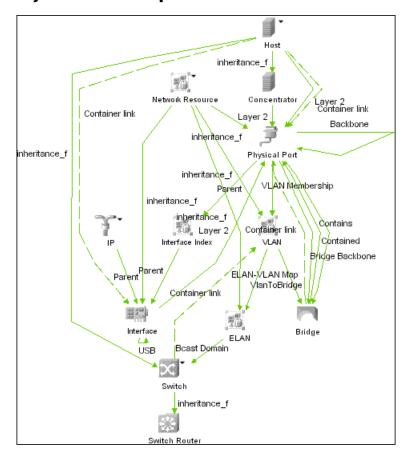
- ➤ The Layer 2 connectivity between the switch-port to the host is based on the host MAC address. These MAC addresses are discovered by the network connection jobs (Host Interfaces).
- ➤ The trigger of the Layer 2 job is dependent on the type of the discovered switch. The switch class and type is discovered by the SNMP connection job.
- ➤ Host Networking by SNMP job. You should run this job on all SNMP agents of the switches that were discovered in the environment. The to-be discovered Layer 2 link names are dependent on this discovery. (Layer 2 link names are the replica of the relevant interface index name and description that the host base pattern discovers.)

3 Discovered CITs

CIT Name	Description
VLAN	Virtual LAN. A logical, not physical, group of devices, defined by software. VLANs enable network administrators to resegment their networks without physically rearranging the devices or network connections.
ELAN	Emulated LAN. A set of clients and servers connected by virtual circuits over an ATM network.
Concentrator	Represents a hub or a switch to which there is no SNMP access.
Bridge	A physical unit inside the switch (like a NIC card) that holds the switch physical ports.
Physical Port	The switch physical port.
Contains	The link between the port and bridge.
VLAN Membership	The link between the port and VLAN.
VlanToBridge	The link between the VLAN and bridge.

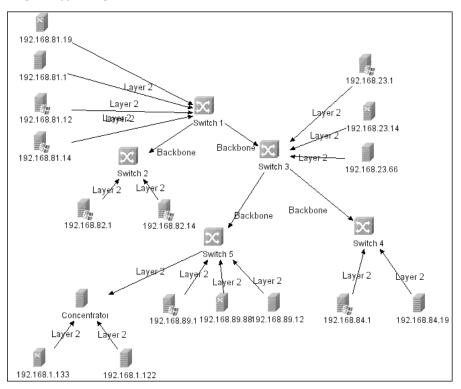
CIT Name	Description
ELAN-VLAN Map	The link between the ELAN and VLAN.
Bcast Domain	The link between the VLAN and switch.

4 Layer 2 Relationships



- ➤ A Layer 2 switch can be connected to its ports directly or through a VLAN.
- ➤ The Bridge CI represents the basic MAC address (Network Interface Card) on which the ports are located.
- ➤ Each switch-port can be connected to a host or interface object (the end user machines) by a Layer 2 link, or to a port-switch by a Backbone link.

5 Topology Map



6 Discovery Workflow

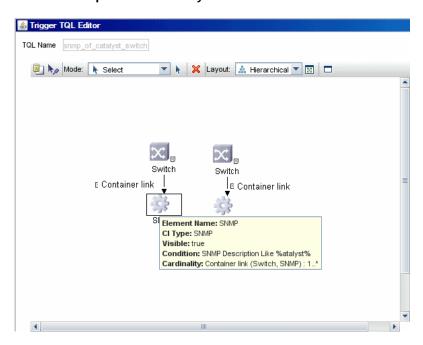
Important: The Layer 2 package includes four jobs. Each job discovers a part of the Layer 2 architecture. You should activate these jobs in the following order.

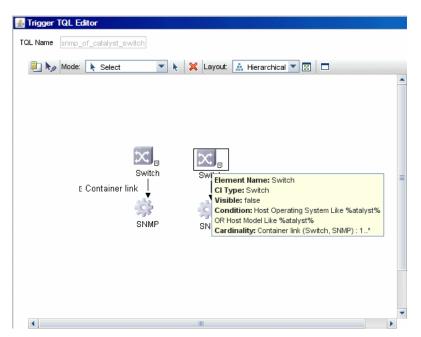
a Activate the **VLANS** by **SNMP** job.

The trigger for this job is the **snmp_of_catalyst_switch** TQL. The Switch CIT is either:

- ➤ an SNMP object that holds a description containing the string atalyst
- ➤ an SNMP agent that is connected to a switch that holds an operating system or model attribute value containing the string atalyst

SNMP Description Like %atalyst%:





Host Operating System Like %atalyst% OR Host Model Like %atalyst%:

The SNMP_Net_Dis_Catalyst_Vlans.py script retrieves the VLAN, ELAN name, and VLAN number per ELAN tables.

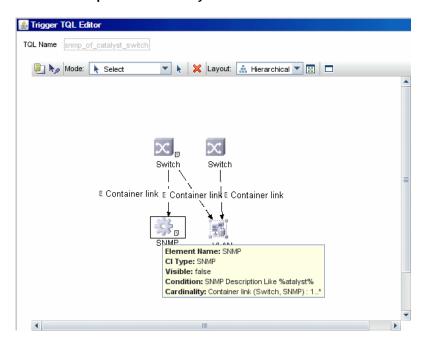
b Activate the **VLAN ports by SNMP** job.

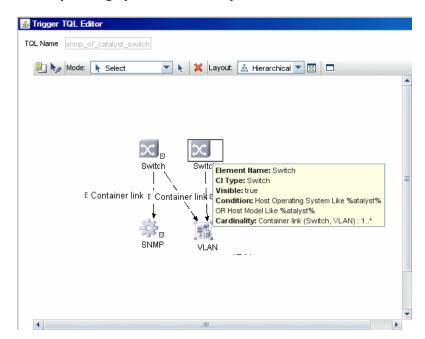
The trigger for this job is the **catalyst_vlan** TQL. This is a VLAN object that has a connection to either:

- ➤ a switch with an SNMP object that holds a description containing the string atalyst
- ➤ a switch that holds an operating system or model attribute value containing the string **atalyst**

The trigger is placed on the VLAN object instead of on the SNMP itself because the VLAN object must be authenticated with a special community string (and not with the regular community string that was discovered on the SNMP object on the discovered switch). This community string should hold the value <COMMUNITY>@<VLAN NUMBER>. For example, if the community string is **public** and the discovered VLAN number is **16**, the community string is **public@16**. For details on the SNMP protocol parameters, see "SNMP Protocol" on page 183.

SNMP Description Like %atalyst%:





Host Operating System Like %atalyst% OR Host Model Like %atalyst%:

The SNMP_Net_Dis_VMS_catalyst.py script retrieves the Base MAC table and Port number If Index table.

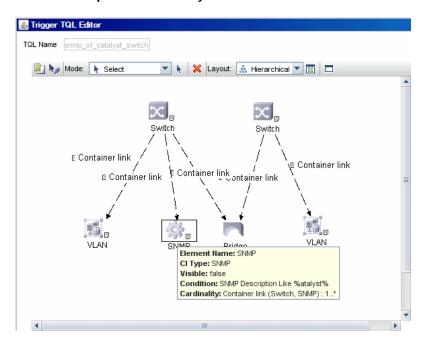
c Activate the Layer2 Topology Bridge based by SNMP job.

The trigger for this job is the **catalyst_bridge_no_vlan** TQL. This is a Bridge object that has a connection to:

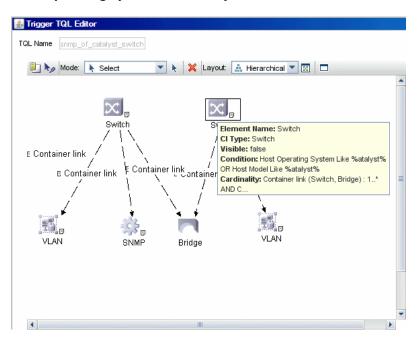
- ➤ a switch with an SNMP object that holds a description containing the string **atalyst**
- ➤ a switch that holds an operating system or model attribute value containing the string **atalyst**.
- ➤ the NOT VLAN Bridge MAC is null attribute

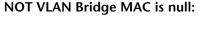
In both the first two cases, the switch should have zero connections to a VLAN.

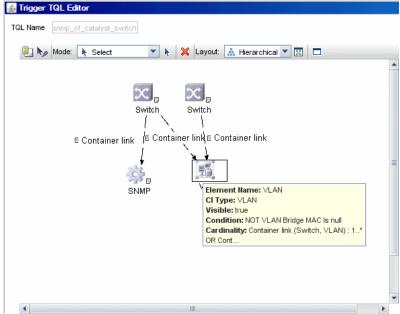
SNMP Description Like %atalyst%:



Host Operating System Like %atalyst% OR Host Model Like %atalyst%:







Both this job (Layer2 Topology Bridge based by SNMP) and the following job (Layer2 Topology VLAN based by SNMP) use the bridgePortDisc.py script. The difference between the jobs in this script is the way they retrieve the community string:

- ➤ Layer2 Topology Bridge based by SNMP uses the regular SNMP community authentication. The job is triggered on the Bridge only when the discovered switch has no VLANS.
- ➤ Layer2 Topology VLAN based by SNMP is triggered on each one of the VLANs discovered on the switch. This job uses the relevant special community authentication, as explained in step b, based on the triggered VLAN number.

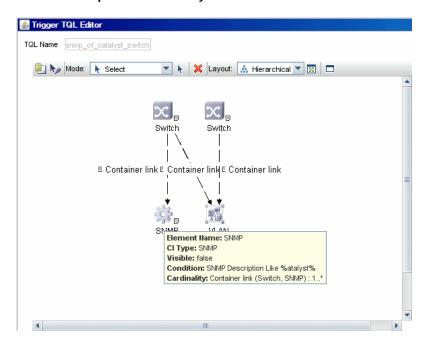
Note:

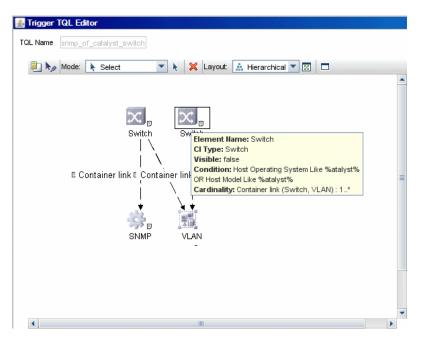
- ➤ When the VLANs by SNMP job runs, it discovers Layer 2 topology that is relevant to the discovered VLAN only.
- ➤ Bridge Layer 2 discovery. If a machine has no VLANs, discovery is triggered on the bridge of the switch. DDM retrieves the Layer 2 topology of all the switches.
- ➤ If you dispatch the Bridge Layer 2 job on the bridge of a switch that holds VLANs only, the default VLAN Layer 2 topology is discovered.
- **d** Activate the Layer2 Topology VLAN based by SNMP job.

The trigger for this job is the **catalyst_vlan_with_bridge** TQL. This is a VLAN object with a value in its bridge_mac attribute. It should also have a connection to either:

- ➤ a switch with an SNMP object that holds a description containing the string atalyst
- ➤ a switch that holds an operating system or model attribute value containing the string **atalyst**

SNMP Description Like %atalyst%:





Host Operating System Like %atalyst% OR Host Model Like %atalyst%:

For details on the bridgePortDisc.py script, see step c.

The Backbone and Layer 2 links are created by the enrichments of the Layer 2 package, based on the data that was discovered by these jobs. After these jobs have run, job statistics do not show any Layer 2 or Backbone links as parts of the results.

7 Troubleshooting and Limitations

- ➤ If the results of the discovery return empty, verify that you have access to the discovered SNMP agent (or to the SNMP agent using the special community authentication) and that all the requested MIB tables are responding to SNMP requests from the Probe machine. For details on the MIB tables, refer to the appropriate script.
- ➤ In cases where the reported bridge MAC address is 0000000000, "", or null, the pattern does not report results.

➤ If the retrieved basic bridge MAC (retrieved from the 1.3.6.1.2.1.17.1.1 table) is not the same as the given bridgeld in the destination data, the pattern returns zero results.

In the case of SNMP_Dis_L2_Bridge, bridgeId is set by bridge_basemacaddr. In the case of SNMP_Dis_L2_VLAN, bridgeId is set by vlan_bridgemac.

🏲 Network – Load Balancer

This task explains how to discover load balancers.

This task includes the following steps:

- ➤ "Overview" on page 335
- ➤ "Supported Versions" on page 336
- ➤ "Prerequisites" on page 336
- ➤ "Load Balancer CITs" on page 337
- ➤ "The Load_balancing Package" on page 338
- ➤ "The Alteon_application_switch Package" on page 339
- ➤ "The F5_BIGIP_LTM Package" on page 340
- ➤ "The Cisco_CSS Package" on page 341
- ➤ "Topology Map" on page 342
- ➤ "Discovery Workflow" on page 342

1 Overview

DDM discovers the following load balancers:

- ➤ F5 BIG-IP Local Traffic Manager (LTM)
- ➤ Nortel Application Switches (formerly known as Alteon Application Switches)
- ➤ Cisco Content Services Switches (CSS)

2 Supported Versions

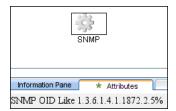
The supported version for each load balancer is as follows:

- ➤ F5 BIG-IP Local Traffic Manager, versions 9 and 4.
- ➤ Nortel Application Switches. No known limitations
- ➤ Cisco Content Services Switches. No known limitations.

3 Prerequisites

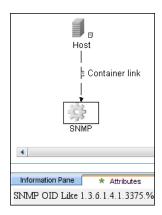
Run the **Host Connection by SNMP** job to discover and create SNMP CIs which answer the following requirements:

➤ To be the trigger TQL for the **Alteon application switch by SNMP** job with the following condition:



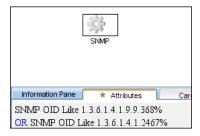
SNMP OID Like 1.3.6.1.4.1.1872.2.5%

➤ To be the trigger TQL for the **F5 BIG-IP LTM by SNMP** job with the following condition:



SNMP OID Like 1.3.6.1.4.1.3375%

➤ To be the trigger TQL for the **Cisco CSS by SNMP** job with the following condition:

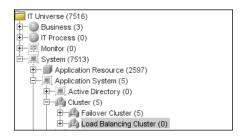


SNMP OID Like 1.3.6.1.4.1.9.9.368% OR 1.3.6.1.4.1.2467%

4 Load Balancer CITs

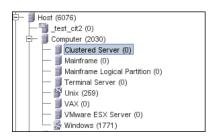
The following CITs model load balancer topology:

Load Balancer Software



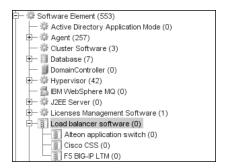
This CIT represents software that provides load balancing solutions. For details on the supported load balancers, see "Overview" on page 335.

Clustered Server



A clustered server is a traffic-management object on the system that can balance traffic load across a pool of servers. Clustered servers increase the availability of resources for processing client requests. The primary function of a clustered server is to receive requests and distribute them to pool members according to criteria you specify.

Load Balancing Cluster



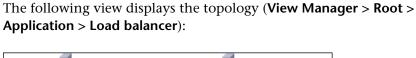
A load balancing cluster (or pool) is a logical set of devices that are grouped together to receive and process traffic. Instead of sending client traffic to the destination IP address specified in the client request, the virtual server sends the request to any of the servers that are members of that pool. This helps to efficiently distribute the load on your server resources.

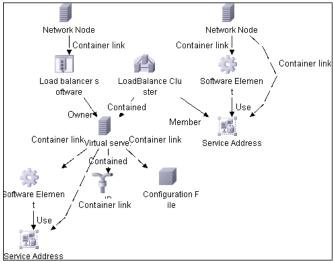
5 The Load_balancing Package

This package contains data that is common to all load balancer solutions.

The Network – Load balancer module contains the load balancing jobs:

- ➤ Alteon application switch by SNMP
- ➤ Cisco CSS by SNMP
- ➤ F5BIG-IP LTM by SNMP





6 The Alteon_application_switch Package

This package contains a class model definition, a pattern, and a job used to discover Nortel application switches by SNMP.

To run this package, activate the **Alteon application switch by SNMP** job. DDM discovers Nortel (Alteon) load balancers and all related CIs.

The following SNMP tables are queried:

Table Name	Name From MIB	OID
Virtual servers	slbCurCfgVirtServerTable	1.3.6.1.4.1.1872.2.5.4.1.1.4.2.1
Virtual services	slbCurCfgVirtServicesTable	1.3.6.1.4.1.1872.2.5.4.1.1.4.5.1
Real groups	slbCurCfgGroupEntry	1.3.6.1.4.1.1872.2.5.4.1.1.3.3.1
Real servers	slbCurCfgRealServerTable	1.3.6.1.4.1.1872.2.5.4.1.1.2.2.1
Port links	slbCurCfgRealServPortTable	1.3.6.1.4.1.1872.2.5.4.1.1.2.5.1
Ports	slbCurCfgPortTable	1.3.6.1.4.1.1872.2.5.4.1.1.5.2.1

7 The F5_BIGIP_LTM Package

This package contains a class model definition, a pattern, and a job used to discover the F5 BIG-IP Local Traffic Manager (LTM) by SNMP. This package supports F5 BIG-IP LTM, versions 4 and 9.

To run this package, activate the **F5 BIG-IP LTM by SNMP** job. DDM chooses all SNMPs related to F5 and runs against them.

The following SNMP tables are queried for version 9:

Table Name	Name From MIB	OID
General information	sysProduct	1.3.6.1.4.1.3375.2.1.4
Virtual servers	ltmVirtualServTable	1.3.6.1.4.1.3375.2.2.10.1.2.1
Pools	ltmPoolTable	1.3.6.1.4.1.3375.2.2.5.1.2.1
Pools to server	ltmVirtualServPoolTable	1.3.6.1.4.1.3375.2.2.10.6.2.1
Pool members	ltmPoolMemberTable	1.3.6.1.4.1.3375.2.2.5.3.2.1
Rules to servers	ltmVirtualServRuleTable	1.3.6.1.4.1.3375.2.2.10.8.2.1
Rules	ltmRuleTable	1.3.6.1.4.1.3375.2.2.8.1.2.1

The following SNMP tables are queried for version 4:

Table Name	Name From MIB	OID
General information	globalAttributes	1.3.6.1.4.1.3375.1.1.1.1
Virtual servers	virtualServerTable	1.3.6.1.4.1.3375.1.1.3.2.1
Pools	poolTable	1.3.6.1.4.1.3375.1.1.7.2.1
Pool members	poolMemberTable	1.3.6.1.4.1.3375.1.1.8.2.1

8 The Cisco_CSS Package

This package contains a class model definition, a pattern, and a job used to discover Cisco Content Serivces Switches by SNMP. This package supports all versions of Cisco CSS.

To run this package, activate the **Cisco CSS by SNMP** job. DDM chooses all SNMPs related to Cisco CSS and runs against them.

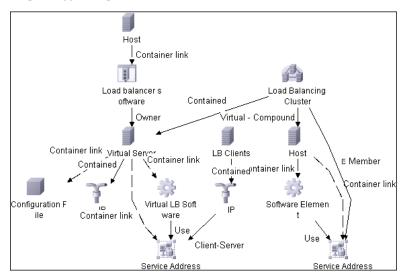
Note: Some services may not be discovered by this package if no content rule is defined for them.

Discovery of CSS is based on three tables: **apCntTable**, **apSvcTable**, and **apCntsvcTable** (see the following table):

- ➤ apCntTable provides information about virtual addresses, virtual services, and pools.
- ➤ apSvcTable provides information about physical hosts included in the pool.
- ➤ apCntsvcTable describes which host is included in which pool.
 apSvcTable can contain entries for which there is no corresponding row in apCntsvcTable. In this case, such hosts are skipped.

Table name	Name from MIB	OID
CNT	apCntTable	1.3.6.1.4.1.2467.1.16.4.1 or 1.3.6.1.4.1.9.9.3681.16.4.1
SVC	apSvcTable	1.3.6.1.4.1.2467.1.15.2.1 or 1.3.6.1.4.1.9.9.3681.15.2.1
CNT to SVC	apCntsvcEntry	1.3.6.1.4.1.2467.1.18.2.1 or 1.3.6.1.4.1.9.9.3681.18.2.1

9 Topology Map



10 Discovery Workflow

- ➤ **Host Connection by SNMP.** For details on the prerequisites to running a load balancer job, see "Prerequisites" on page 336.
- ➤ Any of the following jobs:
 - ➤ F5 BIG-IP Local Traffic Manager (LTM)
 - ➤ Nortel Application Switches (formerly known as Alteon Application Switches)
 - ➤ Cisco Content Services Switches (CSS)

Network Connections – Active Discovery

This task describes how to discover processes.

This task includes the following steps:

- ➤ "Overview" on page 343
- ➤ "TCP Discovery" on page 343
- ➤ "Job Order and Scheduling" on page 344

- ➤ "Supported Versions" on page 344
- ➤ "Discovery Pattern Parameters" on page 344
- ➤ "Network and Protocols" on page 345
- ➤ "Prerequisites" on page 345
- ➤ "The Collect Network Data by Shell or SNMP Job" on page 346
- ➤ "The Servers by Network Data Job" on page 346
- ➤ "The IP Traffic by Network Data Job" on page 347
- ➤ "The Server Ports by Network Data Job" on page 347
- ➤ "The Potential Servers by Network Data Job" on page 348
- ➤ "The Process To Process by Network Data Job" on page 350
- ➤ "Discovered CITs" on page 351
- ➤ "Discovery Workflow" on page 351

1 Overview

This module uses both TCP discovery and process to process discovery to map network connections.

The DDM Probe includes a built-in MySQL database so there is no need to install a separate MySQL instance for either the NetFlow or TCP discovery. Instead, data is saved to a dedicated scheme (called netflow for historical reasons).

2 TCP Discovery

TCP discovery consists of the following separate parts:

- ➤ Gathering the connectivity information (either via running netstat commands or by listening to NetFlow packets). This part is achieved by running the following jobs: Collect Network Data by Shell or SNMP and Collect Network Data by NetFlow.
- ➤ Querying the gathered network data through dedicated DDM jobs.

3 Job Order and Scheduling

By default, all queries are scheduled to be run on a relatively frequent basis (every hour). The queries themselves are not re-run unless the data set has changed since the last run, in order not to waste CPU cycles on the Probe.

Although you can activate the Collect Network Data by Shell or SNMP job together with the relevant queries, you would probably not see any results until at least one hour has passed before the next scheduled invocation of the query. This is because, by the time the first set of queries is run, no data has yet been gathered. So a best practice is to make sure data gathering is complete and only then launch the query and see the result it populates.

4 Supported Versions

DDM supports NetFlow versions 5 and 7.

5 Discovery Pattern Parameters

You can filter the list of processes to run only those processes that retrieve data that is of interest to you. The network connectivity of these processes is omitted.

To view the list of processes that are filtered by default:

- **1** Select the Manage Discovery Resources window.
- **2** Select the **TCP Discovery** package.
- **3** Select the pattern.
- **4** Display the **Pattern Signature** tab and locate the **Discovery Pattern Parameters** pane.

The values are displayed in the **Value** box and are comma-separated. For details on adding, deleting, or editing parameters, see "Discovery Pattern Parameters Pane" on page 236.

To override a process parameter value:

- 1 Select Run Discovery (Advanced Mode) > Network Connections Active Discovery.
- **2** Select the job, select the **Properties** tab, and locate the **Parameters** pane. For details on overriding a parameter, see "Parameters Pane" on page 149.

6 Network and Protocols

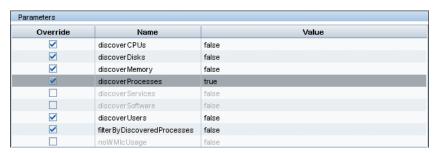
To discover network connections, define the following protocols:

- ➤ NTCmd. For details, see "NTCMD Protocol" on page 181.
- ➤ **SNMP**. For details, see "SNMP Protocol" on page 183.
- ➤ **SSH.** For details, see "SSH Protocol" on page 186.
- ➤ **Telnet**. For details, see "Telnet Protocol" on page 188.
- ➤ WMI. For details, see "WMI Protocol" on page 194.

Note: None of these protocols is mandatory, but WMI alone does not retrieve network data.

7 Prerequisites

- **a** Run the following jobs in the **Network Basic** module:
 - ➤ Range IPs by ICMP job
 - ➤ Host Connection by Shell/SNMP/WMI (NTCmd, SSH, Telnet, and WMI CIs are discovered)
- **b** Run Host Resources By Shell/SNMP/WMI in the Network Host Resources module.
- **c** For the **Process To Process by Network Data** job, set **discoveryProcesses** to **true**, since that is the only required parameter, and all other parameters to **false**:



Important: Host Connection by Shell/SNMP/WMI and Host Resources By Shell/SNMP/WMI must run within the same time range, so that the process IDs are mapped to the ports on which the processes are listening.

8 The Collect Network Data by Shell or SNMP Job

This is the main job. It discovers the TCP connections of the discovered machines using Shell or SNMP, and stores the information in the probededicated netflow database.

The following TCP jobs query the data stored by this job. These jobs are scheduled to run every hour: they query the database to verify whether new data exists that was discovered by the Collect Network Data by Shell or SNMP job.

9 The Servers by Network Data Job

This job enables the discovery of specific service names (the services parameters). The service name to port numbers are still configurable through the portNumberToPortName.xml file.

The links discovered by this job are clientserver links (between the client IP and the server port to which it connects) and dependency links between the related hosts.

You can override the values of the following pattern parameters:



- ➤ ignoreLocalConnections. DDM should ignore local connections. The default is false.
- ignoreUnackedTcpConn. DDM does not report unacknowledged connections.
- ➤ ignoredlps. IPs that should be filtered. The values are comma separated (for example, 10.*.*.*,15.45.*.*). The default is **none** (that is, the value is empty).
- ➤ includeOutscopeClients/Services. Prevents discovery of clients or services on machines that are out of a Probe's network scope.
- ➤ onlyHostDependLinks. Enables discovery of dependency links only (without the clientserver links).
- ➤ services. The following default services are discovered: http, https, telnet, ssh, rmi, websphere, weblogic, weblogicSSL, oracle, sql, netflow, siebel, and sap. This parameter can also include specific numbers and an asterisk to represent all known ports.
- ➤ updateUtilizationInfo. Relevant only for NetFlow and can be used to prevent reporting the packets and octets count information on the clientserver links.

10 The IP Traffic by Network Data Job

This job discovers traffic links between all communicating IPs. The traffic links are populated between any two IPs that are seen to communicate. On these links is an attribute with the value of the top ports (the most important TCP/UDP ports) that were found between those two IPs/hosts. The top ports are calculated according to the number of clients and the size of the network traffic between them.

You can configure the maximum number of recognized, interesting ports through the maxPorts parameter.

11 The Server Ports by Network Data Job

This job discovers open server ports according to a list of specified services. This can be useful if you do not want to discover the TCP connections themselves but do want to know which ports are open, without performing any TCP port scanning (which may be dangerous in some organizations).

This discovery job is not relevant for NetFlow data as there is no LISTEN flag in this case.

You can override the values of the following pattern parameters:



- ➤ includeOutscopeServers. Prevents discovery of servers on machines that are out of a Probe's network scope.
- ➤ services. The following default services are discovered: http, https, telnet, ssh, rmi, websphere, weblogic, weblogicSSL, oracle, sql, netflow, siebel, and sap. This parameter can also include specific numbers and an asterisk to represent all ports.

12 The Potential Servers by Network Data Job

This job can be used in situations where you need to find clientserver links but without defining the port numbers in advance. The server port is defined according to the criteria passed as job parameters: minClient (the minimum number of clients for the service), minPackets/minOctets (minimum packets and octets – relevant only for NetFlow).

You can override the values of the following pattern parameters:



- ➤ disregardListenPorts. False: DDM checks whether the port is marked as a listening port. If it is, DDM does not check the minimal conditions (minOctets, minClients, and minPackets). True: DDM does not check if the port is a listening port and instead uses the minimal conditions (minOctets, minClients, and minPackets). The default is false.
- ➤ ignoredlps. IPs that should be filtered. The values are comma separated (for example, 10.*.*.*,15.45.*.*). The default is **none** (that is, the value is empty).
- ➤ includeOutscopeClients/Services. Prevents discovery of clients or services on machines that are out of a Probe's network scope.
- ➤ ignoreLocalConnections: Local connections should not be discovered. The default is **false**.
- ➤ minClients. The number of connected clients that must be discovered to make this service a potential server.
- ➤ minOctets. The number of octets (bytes) to be sent by a client to a service, so that the client is included in the discovery.
- ➤ minPackets. The number of packets to be sent by a client to a service, so that the client is included in the discovery.
- ➤ protocols. Limit the query to these IP protocols. The values are comma separated. The default is **TCP**.

Important: This job does not come out-of-the-box with a Trigger TQL because it is not intended to be used on many triggers. Rather, you should activate the job manually against specific IP instances, to find unknown server ports. It is preferable to add the ports afterwards to the portNumberToPortName.xml file and continue discovery through the **Servers by Network Data**.

13 The Process To Process by Network Data Job

This job discovers processes running on Windows 2003 and Windows XP, SunOS, Hewlett Packard UniX (HP-UX), AIX, and Linux operating systems. For the HP-UX and AIX machines, you should install lsof software, which can be downloaded from the Internet from, for example,

http://www.netadmintools.com/html/lsof.man.html. You can install lsof software also on SunOs. If you do not, the pfiles software that is installed on SunOS is used.

For details on network protocols that must be defined, see "Network and Protocols" on page 345.

You can override the values of the following pattern parameters:



- ➤ filterProcessesByName. Limits the processes to be run according to the names in the Value box. The default is system, svchost.exe, and Isass.exe.
- ➤ ignoreLocalConnections: Local connections should not be discovered. The default is **false**.
- ➤ reportPortRelationShips. Extra relationship links (for example, Client-Server and use) for the server port should be reported.
 - By default, Client-Server links between processes and service addresses are reported. **True**: Client-Server links between client IPs and service addresses and Client-Server links betweeen two processes are also reported.
- ➤ services. Specifies which services should be discovered. The default is all (*).

14 Discovered CITs

- ➤ Client-Server. DDM determines which machine is the server and which the client:
 - ➤ If one of the sites discovered as a listening port, then this side is supposed to be a server.
 - ➤ if one side has more than 2 connections on its ports, it supposed to be the server side
 - ➤ If both sides have just one connection to a port, DDM identifies whether it is a server, by checking the ports and the portNumberToPortName.xml file (Manage Discovery Resources > Discovery Resources > Network > Configuration Files).
 - ➤ otherwise, the port is checked whether it is equals, or is less than, **1024**, DDM identifies it as a server.
- ➤ Talk. This link is created between two processes only if DDM does not recognize the Client-Server link between the processes. The Talk link reports bidirectionally.

15 Discovery Workflow

In the Run Discovery window, activate the jobs in the following order:

- ➤ "Network Host Resources" on page 315
- ➤ "The Collect Network Data by Shell or SNMP Job" on page 346
- ➤ "The Servers by Network Data Job" on page 346
- ➤ "The IP Traffic by Network Data Job" on page 347
- ➤ "The Server Ports by Network Data Job" on page 347
- ➤ "The Potential Servers by Network Data Job" on page 348
- ➤ "The Process To Process by Network Data Job" on page 350

🏲 Virtualization – VMware

This task describes how to discover the VMware Infrastructure Topology suite of applications.

This task includes the following steps:

- ➤ "Overview" on page 353
- ➤ "Supported Protocol Versions" on page 353
- ➤ "Supported VMware Servers" on page 353
- ➤ "SSL Support Details" on page 353
- ➤ "Prerequisites Add JAR Files" on page 354
- ➤ "Prerequisites Run Host Discovery" on page 354
- ➤ "Prerequisites Run WMI Discovery" on page 354
- ➤ "Prerequisites Run Processes Discovery" on page 355
- ➤ "Prerequisites VMware Infrastructure Permissions" on page 355
- ➤ "Network and Protocols" on page 355
- ➤ "Virtual Topology Map" on page 357
- ➤ "Licensing Topology Map" on page 357
- ➤ "Discovery Workflow Overview" on page 358
- ➤ "Discovery Workflow VMware VirtualCenter Connection by WMI and VIM" on page 358
- ➤ "Discovery Workflow VMware VirtualCenter Topology by VIM" on page 361
- ➤ "Discovery Workflow VMware ESX Connection by VIM" on page 364
- ➤ "Discovery Workflow VMware ESX Topology by VIM" on page 366
- ➤ "Troubleshooting and Limitations" on page 369

1 Overview

The VMware discovery process enables you to discover virtual machines (VM), processors, memory, storage, and network resources that are running on VMware.

2 Supported Protocol Versions

There are two protocol versions available: 2.0 and 2.5. The new versions of the ESX servers support the VMware Infrastructure SDK API, version 2.5 but transparently support connections using the old version of the protocol, providing backward compatibility. Older versions of the servers support the VMware Infrastructure SDK API, version 2.0 only. For details, see the next section.

For details on the protocol, see "VMware Infrastructure Management Protocol (VIM)" on page 190.

3 Supported VMware Servers

- ➤ ESX Server 3.5, VirtualCenter Server 2.5, and ESX Server 3i support VMware Infrastructure SDK API 2.5 (they can be connected using protocol version 2.5 or 2.0)
- ➤ ESX Server 3.0.x, VirtualCenter Server 2.0.x support VMware Infrastructure SDK API 2.0 (they can be connected using protocol version 2.0 only)

4 SSL Support Details

Web services use http transport which can also be transferred over SSL. The VMware Infrastructure Management (VIM) protocol uses SSL by default, but it is possible to configure it without SSL usage.

Each server supporting the VIM protocol (VirtualCenter server or ESX server) has its own SSL certificated.

Currently, DDM supports only one strategy (accept all certificates always). The following code is an example of how DDM sets the global property for the Axis engine:

System.setProperty('org.apache.axis.components.net.SecureSocketFactory', 'org.apache.axis.components.net.SunFakeTrustSocketFactory')

5 Prerequisites - Add JAR Files

To use the VMware Infrastructure Management protocol, add the following JAR files from the SDK to the Probe:

- vim.jar. Contains Java classes generated by Axis from WSDL for API version 2.0
- ➤ vim25.jar. Contains Java classes generated by Axis from WSDL for API version 2.5

These JAR files are used without any modification together with the Axis engine. All protocol interactions are performed by working with objects from these jar files (instantiating objects, calling methods, getting result objects, and so on).

Note: These JAR files are not included by default with DDM due to licensing issues.

- **a** Download the VMware Infrastructure SDK from **http://www.vmware.com/support/developer/vc-sdk/**, version 2.5.0.
- **b** Locate the vim.jar and vim25.jar files in the SDK\samples\Axis\java directory.
- **c** Copy the JAR files to the **C**:\hp\DDM\DiscoveryProbe\root\ext\vmware directory.
- **d** To load the JAR files, restart the Probe.

6 Prerequisites - Run Host Discovery

To be able to connect to each potential VMware server (VirtualCenter or ESX), discover its Host CI by running one of the **Host Connection by Shell/WMI/SNMP** jobs (in the Network – Basic module).

7 Prerequisites – Run WMI Discovery

To be able to connect to each potential VirtualCenter server (this is not required for ESX), make the WMI connection available for the host by running the **Host Connection by WMI** job.

8 Prerequisites – Run Processes Discovery

To be able to connect to each potential VMware server (VirtualCenter or ESX), you must discover Process CIs that match certain criteria, by running one of the **Host Resources by Shell/WMI/SNMP** jobs (in the Network – Basic module).

9 Prerequisites – VMware Infrastructure Permissions

VMware Infrastructure discovery requires special permissions for protocol used.

The VMware Infrastructure Management (VIM) protocol requires the following permissions:

- ➤ **System.Read** permissions for users performing login (for the Read-Only user group) for all discovery.
- ➤ Global.Licenses permissions to obtain the total and available number of licenses for each License Feature. If the user does not have these permissions, these attributes remain empty.

The WMI protocol used in the VirtualCenter connection pattern requires the following permissions:

➤ Users should be able to perform remote queries for the **root\default** namespace (**Remote Enable**, **Enable Account**, and **Execute Methods**); administrators usually have these permissions.

10 Network and Protocols

The WMI, Shell (Telnet, SSH, NTCMD), and SNMP protocols are required to discover hosts and host processes.

The WMI protocol is required to discover the VirtualCenter connectivity pattern.

The VMware Infrastructure Management (VIM) protocol is required for all VMware jobs.

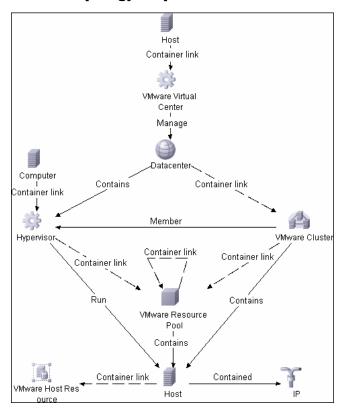
a Set up the **WMI protocol**. For details, see "WMI Protocol" on page 194. The protocol requires the user name and password and, optionally, the domain name.

- **b** Set up the **Shell protocol**. For details, see "Telnet Protocol" on page 188, "SSH Protocol" on page 186, or "NTCMD Protocol" on page 181. These protocols require the user name, password, and domain name (the domain name is optional for NTCMD).
- **c** Set up the **SNMP protocol**. For details, see "SNMP Protocol" on page 183. This protocol requires the community name (for v2), the user name (for v3), and the password (for v3).
- **d** Set up the **VMware Infrastructure Management (VIM)** protocol. For details, see "VMware Infrastructure Management Protocol (VIM)" on page 190. This protocol requires a user name and password.

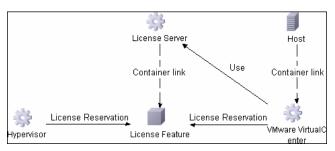
Port Number is optional.

Use SSL. true: select if the VMware servers are configured to use SSL by default. **false**: select if the VMware servers are configured to use non-secured http.

11 Virtual Topology Map



12 Licensing Topology Map



13 Discovery Workflow - Overview

The Network – VMware module includes two jobs for VirtualCenter discovery and two for ESX server discovery:

- ➤ If the VMware Infrastructure environment is managed by VirtualCenter Servers, run the VMware VirtualCenter Connection by WMI and VIM job, followed by the VMware VirtualCenter Topology by VIM job.
- ➤ If the VMware Infrastructure environment includes unmanaged ESX servers (standalone) or the entire environment is unmanaged, run the VMware ESX Connection by VIM job, followed by the VMware ESX Topology by VIM job.

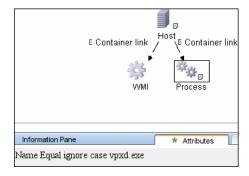
Note: The **Manual VMware VIM Connection** job is intended for use in those instances when the above four jobs cannot discover the VMware environment. You must, however, manually run this job, that is, you specify a URL (you need to know its format), you activate the job, and you choose the Probe.

14 Discovery Workflow – VMware VirtualCenter Connection by WMI and VIM

This job discovers VirtualCenter Servers.

Trigger CI. WMI.

Trigger TQL and Node Conditions:



Triggered CI Data:

- ➤ credentialsId. The credentials ID of the WMI agent CI.
- ➤ ip_address. The IP address, taken from the WMI agent CI.

 Discovery Pattern Parameters. None.

Discovery and Dependency Mapping performs the following processes:

- ➤ Runs through all defined credentials for the VMware Infrastructure Management (VIM) protocol.
- ➤ If the **Use SSL** parameter is set to **true**, the default prefix is HTTPS, otherwise the prefix is set to HTTP.
- ➤ If the user has entered a port number in the VIM protocol, this value is used for the port. If not, a WMI query is performed to extract the port number from the registry. DDM queries HKLM\SOFTWARE\VMware, Inc.\VMware VirtualCenter and searches for the HttpsProxyPort or HttpProxyPort attribute.
 - ➤ If the HttpsProxyPort attribute is found, DDM uses its value for the port and sets the prefix to HTTPS.
 - ➤ If the HttpProxyPort attribute is found, DDM uses its value for the port and sets the prefix to HTTP.

Note: DDM performs a search for the WMI port once only. The retrieved value is cached so that the same query does not need to be run for each VMware Infrastructure Management (VIM) protocol entry.

- ➤ DDM creates a VMware Infrastructure client, passes the user name and password from the current VMware Infrastructure Management (VIM) protocol, passes the generated URL, and performs a connection.

The connection is made using the version 2.5 protocol. If this connection fails, DDM tries to connect using the version 2.0 protocol.

- ➤ If the connection is successful, DDM retrieves the product information and extracts the required values (these values are stored in the VMware VirtualCenter CI attributes). The values include build number, version, description, and so on.
- ➤ DDM uses the IP address to create a Host CI.
- ➤ DDM stores the generated URL used for this successful connection in the VirtualCenter CI's **connection_url** attribute.
- ➤ DDM stores the **credentialsId** of the current VIM protocol in the VirtualCenter CI's **credentialsId** attribute.
- ➤ If the connection is successful, DDM clears all errors and warnings that were generated in previous connection attempts and returns results.
- ➤ If the connection is unsuccessful, DDM continues with the next VIM protocol credentials entry, until all are tried.

Output:



Troubleshooting:

➤ **Problem.** The following error message is displayed when an operation cannot be performed due to lack of permissions:

User does not have required '<permission>' permission

Check that permissions are set as **System.Read**.

➤ **Problem.** The following error message is displayed when credentials are not correct:

Invalid user name or password

15 Discovery Workflow - VMware VirtualCenter Topology by VIM

This job connects to VirtualCenter Servers and discovers the full VMware Infrastructure topology.

Trigger CI. WMI.

Trigger TQL:



Node Conditions. None.

Triggered CI Data:

- ➤ **credentialsId**. The credentials ID of the VMware Infrastructure Management (VIM) protocol saved in the VirtualCenter Server's attribute.
- ➤ **server_url**. The URL for connecting to VMware Infrastructure, taken from the VirtualCenter Server's **connection url** attribute.

Discovery Pattern Parameters. reportPoweredOffVMs. Checks whether virtual machines that are powered off should be reported.

Discovery and Dependency Mapping performs the following processes:

- **1** DDM extracts the connection URL and the VIM protocol credentials ID by using the VirtualCenter Trigger CI. DDM uses the credentials ID to retrieve the user name and password for the VIM protocol. DDM creates a VMware Infrastructure client and connects to the server using these parameters.
 - The connection is made using the version 2.5 protocol. If this connection fails, DDM tries to connect using the version 2.0 protocol.
- **2** DDM performs a query to retrieve information about Datacenters; the retrieved information is used to create Datacenter CIs.
- **3** DDM performs a query for the licensing information, including license availability and usage information, and information about license sources. The user used to retrieve availability information must have **Global.Licenses** permissions. If these permissions do not exist, DDM cannot add the **licenses_total** and **licenses_available** attributes for each License Feature CI, and a warning is reported.

- **4** For each Datacenter, DDM performs a query to retrieve **ComputeResources** data. ComputeResource can represent either a single ESX server or a cluster (in which case it is called ClusterComputeResource). DDM does not map the ComputeResource resource itself to any CI (it is considered an abstract element of the hierarchy) but does use its properties.
- **5** For each ComputeResource resource which is a ClusterComputeResource resource, DDM treats the resource as a cluster and creates a Cluster CI. DDM performs an additional query to retrieve its attributes.
- **6** For each ComputeResource resource, DDM performs queries to retrieve:
 - ➤ Information about its resource pools (the hierarchy of all the resource pools are retrieved in one query).
 - ➤ Information about its ESX servers (all ESX servers are returned in one query; for a ComputeResource resource that is not a cluster, a single ESX is returned).
 - ➤ Information about its VMs (all in one query).
- **7** For each ESX server, DDM discovers its licensing information. For details, see step 3 on page 361.
- **8** When discovering VMs:
 - ➤ DDM retrieves the host key for the Network Node CI, representing the guest OS, which can be either the lowest MAC address or the IP address. To make this information available, the VM must have a VMware Tools component installed and running. If this component is not installed, DDM reports a warning and skips that VM.
 - ➤ If the Tools component is installed, DDM tries to retrieve the host key. DDM searches for the lowest MAC address, or, if that is not available, for the IP. If that is also not available, DDM skips this VM and reports a warning.
 - ➤ DDM determines the power status of the VM: If it is powered-off, the reportPoweredOffVms parameter determines whether DDM skips the machine or includes it in the results. (You may not want to report a powered-off VM because the information it contains—for example, the IP address—may be outdated and may conflict with another VM that is powered-on.

If **reportPoweredOffVms** is set to **false**, the powered-off VM is not reported.

If **reportPoweredOffVms** is set to **true**, DDM tries to include the VM in the results (see the next step).

➤ All discovered VMs undergo a filtering mechanism. Currently filtering is performed by host keys. If there are two machines with the same host key, DDM reports only one, as follows:

If both machines are powered-on, DDM reports the first that is found.

If both machines are powered-off, DDM reports the first that is found.

If the machines have different power states, DDM reports the poweredon machine.

9 All retrieved information is processed: DDM organizes the resource pools into a hierarchy and aligns each VM to its corresponding pool, then creates corresponding CIs and links, and returns the results.

Output:



Troubleshooting:

➤ **Problem.** The following error message is displayed when an operation cannot be performed due to lack of permissions:

User does not have required '<permission>' permission

Check that permissions are set as **System.Read**.

➤ **Problem.** The following error message is displayed when credentials are not correct:

Invalid user name or password

➤ **Problem.** The following warning message is displayed and the CI is not reported:

Cannot determine the IP or MAC address of virtual machine '<vm_name>

➤ **Problem.** The following warning message is displayed, the status is <status> and the CI is not reported:

Virtual machine '<vm_name>' does not have a VMware Tools running

➤ **Problem.** The following warning message is displayed when DDM cannot retrieve license availability (permissions, in most cases, is **Global.Licenses**):

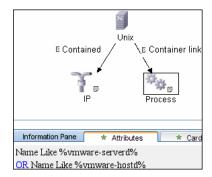
User does not have required '<permission>' permission, features availability information won't be reported

16 Discovery Workflow - VMware ESX Connection by VIM

This job discovers the connections to VMware ESX servers.

Trigger CI. Unix.

Trigger TQL and Node Conditions:



Triggered CI Data. **ip_address**. The IP address is taken from the WMI agent CI.

Discovery Pattern parameters. None.

Discovery and Dependency Mapping performs the following processes:

- ➤ DDM checks the credentials for the VIM protocol.
- ➤ If the current credential includes a defined port, DDM uses this port.

 Otherwise, the port is not specified in the generated connection URL.

 The prefix is determined from the current credential's use SSL attribute.
- ➤ DDM generates a connection URL: cprefix>://<ip_address>:<port>/sdk.
- ➤ DDM creates a VMware Infrastructure client and connects using the generated URL and the user name and password from the credentials.
 - The connection is made using the version 2.5 protocol. If this connection fails, DDM tries to connect using the version 2.0 protocol.
- ➤ If the connection is successful, DDM obtains the product details for the ESX server (version, build, and description), which will be used to populate the attributes of the **Virtualization Layer Software** CI.
 - In addition, DDM retrieves the UUID and name of the ESX server. ESX UUID is stored in the host_key attribute of the VMware ESX Server CI, which is a key attribute.

Chapter 8 • Discovery and Dependency Mapping Content

➤ DDM clears all errors or warnings and returns all discovered results.

Otherwise, if the connection was unsuccessful, DDM tries the next VIM protocol credentials, until all are tried.

Output:



Troubleshooting:

➤ **Problem.** The following error message is displayed when an operation cannot be performed due to lack of permissions:

User does not have required '<permission>' permission

Check that permissions are set as System.Read.

➤ **Problem.** The following error message is displayed when credentials are not correct:

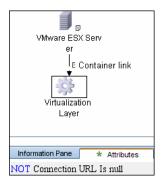
Invalid user name or password

17 Discovery Workflow - VMware ESX Topology by VIM

This job connects to ESX servers and discovers their topology.

Trigger CI. Virtualization Layer Software.

Trigger TQL and Node Conditions:



Triggered CI Data.

- ➤ **credentialsId**. The credentials ID of the VMware Infrastructure (VIM) protocol, saved in the ESX server attribute.
- ➤ server_url. The URL for connection, taken from the ESX server connection url attribute.

Discovery Pattern Parameters. reportPoweredOffVMs. Checks whether VMs that are powered off should be reported.

Discovery and Dependency Mapping performs the following processes:

- ➤ DDM uses the connection URL (extracted from the ESX server attribute) and the user name and password (obtained by the credentialsId Trigger CI from the ESX server attribute) to connect to the server.
 - The connection is made using the version 2.5 protocol. If this connection fails, DDM tries to connect using the version 2.0 protocol.
- ➤ DDM performs discovery of the ESX servers. DDM uses the same objects as the VMware VirtualCenter Topology by VIM job, so the flow is identical. (For details, see "Discovery Workflow VMware VirtualCenter Topology by VIM" on page 361.)

DDM discovers:

- ➤ All resource pools of the server
- ➤ All virtual machines of the server

- ➤ DDM performs discovery of the licensing information (as in the VMware VirtualCenter Topology by VIM job).
- ➤ DDM processes and returns results.

Output:



Troubleshooting:

➤ **Problem.** The following error message is displayed when an operation cannot be performed due to lack of permissions:

User does not have required '<permission>' permission

Check that permissions are set as **System.Read**.

➤ **Problem.** The following error message is displayed when credentials are not correct:

Invalid user name or password

➤ **Problem.** The following warning message is displayed and the CI is not reported:

Cannot determine the IP or MAC address of virtual machine '<vm name>

➤ **Problem.** The following warning message is displayed, the status is <status> and the CI is not reported:

Virtual machine '<vm name>' does not have a VMware Tools running

➤ **Problem.** The following warning message is displayed when DDM cannot retrieve license availability (permissions, in most cases, is **Global.Licenses**):

User does not have required '<permission>' permission, features availability information won't be reported

18 Troubleshooting and Limitations

➤ **Problem.** The following error message is displayed:

ImportError: No module named vmware

Cause. The SDK jars are not copied to the Probe.

Solution. Copy the JAR files to the Probe, as described in "Prerequisites – Add JAR Files" on page 354.

➤ **Problem.** The following error message is displayed:

User does not have required 'System.Read' permission

Cause. There is a lack of permissions from the user account when DDM connects to the ESX server's VirtualCenter.

Solution.

- ➤ Verify that credentials are defined for the VMware Infrastructure Management (VIM) protocol in the proper priority, so that credentials with full permissions have a lower index number than credentials with less permissions. For details, see "Index" on page 179.
- ➤ If DDM previously discovered connections using credentials with less than full permissions, you must rerun the connection job (either VMware VirtualCenter Connection by WMI and VIM or VMware ESX Connection by VIM) to update the credentials ID attribute of VirtualCenter or ESX server, and then run the topology job (VMware VirtualCenter Topology by VIM or VMware ESX Topology by VIM).

- ➤ Currently if VMware Tools are not running on a virtual machine, it is not possible to get its IP or MAC address; such virtual machines are ignored and are not reported to HP Universal CMDB.
- ➤ DDM can discover the total number of licenses and available licenses for each feature, but only when the user has **Global.Licenses** permission. If the user does not have such permissions, these attributes of the License Feature CI are not populated.
- ➤ Different versions of ESX Servers (versions 3.0 and 3.5) report the feature_is_edition flag differently for the esxFull feature: for the older version it is reported as false and for the newer version it is reported as true. Because of this discrepancy, DDM does not report this attribute.
- ➤ Different versions of ESX Servers (versions 3.0 and 3.5) report the total or available license counts differently for ESX-specific features (nas, iscsi, vsmp, san) that are included in the esxFull edition license. For these features, DDM does not report these attributes.
- ➤ There is a difference between VMware protocols 2.5 and 2.5 some attributes appear only in version 2.5 and do not appear in previous version. As the result when using old protocol some attributes won't be discovered, in particular for clusters and licenses.

Web Servers – IIS

This task describes how to discover Internet Information Services (IIS).

This task includes the following steps:

- ➤ "Supported Versions" on page 370
- ➤ "Network and Protocols" on page 371
- ➤ "Discovered CITs" on page 371
- ➤ "Discovery Workflow" on page 371

1 Supported Versions

IIS versions 5 and 6.

2 Network and Protocols

Set up the **NTCMD protocol** and verify that the target machine running IIS lies in the Probe range. For details, see "NTCMD Protocol" on page 181.

3 Discovered CITs



The dependency list for the IIS Web Site node is defined as follows (for details, see "Subgraph Condition Definition Dialog Box" in *Model Management*):



4 Discovery Workflow

In the Run Discovery window, activate the jobs in the following order:

- ➤ WebServices by URL
- ➤ IIS Applications by NTCMD

Chapter 8 • Discovery and Dependency Mapping Content

Importing Data from External Sources

This chapter provides information on the methods of importing data from external sources (CSV files, properties files, and databases).

This chapter includes:

Concepts

- ➤ Importing Data from External Sources Overview on page 374
- ➤ The External_source_import Package on page 376
- ➤ The Import from CSV File Job on page 378
- ➤ The Import from Properties File Job on page 380
- ➤ The Import from Database Job on page 381
- ➤ The External Source Mapping Files on page 386
- ➤ Converters on page 386

Tasks

➤ Import CSV Data from an External Source – Scenario on page 389

Troubleshooting and Limitations on page 393

\lambda Importing Data from External Sources – Overview

Your data is probably stored in several formats, for example, in spreadsheets, databases, XML documents, properties files, and so on. You can import this information into HP Universal CMDB and use the UCMDB functionality to model the data and work with it. External data are mapped to CIs in the UCMDB.

The following external data sources are currently supported:

- ➤ "Comma Separated Value (CSV) Files" on page 374
- ➤ "Properties Files" on page 375
- ➤ "Databases" on page 376

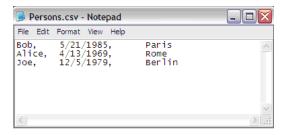
Comma Separated Value (CSV) Files

A *.csv file has a format that stores tabular data. Each row in a CSV file represents a set of values delimited with a particular delimiter. All rows are homogeneous, that is, each row has the same number of values. Values from all rows with the same index create a column. Values in a single column represent the same type of data. Therefore a CSV file represents a table of data (with rows and columns).

The default delimiter for CSV files is the comma, but any symbol can be used as a CSV delimiter, for example, a horizontal tab.

Note: Microsoft Office Excel includes native support for the CSV format: Excel spreadsheets can be saved to a CSV file and their data can then be imported into the UCMDB. CSV files can be opened in an Excel spreadsheet.

Example of a CSV file



CSV Files with Column Titles in First Row

CSV files often include column headings in the first row. When data is imported from these files, the titles are considered data and a CI is created for this row. To prevent a CI being created, you can define which row DDM should start at when importing data from a CSV file:

- 1 Select Manage Discovery Resources > Discovery Resources pane > Discovery Packages > External_source_import package > Patterns > Import_CSV.
- **2** In the Pattern Signature tab, locate the **Discovery Pattern Parameters** pane.
- **3** Locate the **rowToStartIndex** parameter.

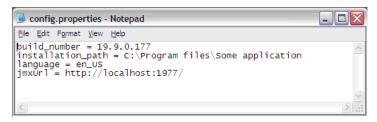
 By default, the value is **1**, that is, DDM retrieves data from the first row.
- **4** Replace **1** with the number of the row at which to start retrieving data. For example, to skip the first row and start with the second row, replace **1** with **2**.

Properties Files

A properties file is a file that stores data in the **key = value** format. Each row in a properties file contains one key-to-value association. In code terms, a properties file represents an associative array and each element of this array (key) is associated with a value.

A properties file is commonly used by an application to hold its configuration. If your application uses a configuration file, you can model the application in the UCMDB.

Example of a properties file



Databases

A database is a widely used enterprise approach to storing data. Relational databases consist of tables and relations between these tables. Data is retrieved from a database by running queries against it.

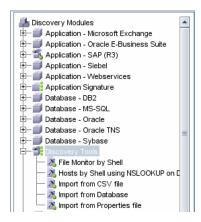
The following databases are supported: Oracle, Microsoft SQL Server, MySQL, and DB2.

The External_source_import Package

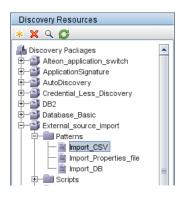
The External_source_import package consists of three jobs and three patterns. There is one job and one pattern for each external source (CSV file, properties file, database):

External Source	Job	Pattern
CSV file	Import from CSV file	Import_CSV
Properties file	Import from Properties file	Import_Properties_file
Database	Import from Database	Import_DB

The jobs are located under the **Discovery Tools** module:



The patterns are located in the **External_source_import** package:



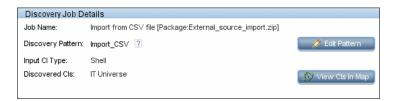
\lambda The Import from CSV File Job

This section includes the following topics:

- ➤ "Job Details" on page 378
- ➤ "Discovery Pattern Parameters" on page 378
- ➤ "Delimiters, Quotes, and Escaping Characters" on page 379

Job Details

The job details are as follows:



This job has no Trigger TQLs associated with it. That is, this job is not triggered automatically (nor are the Import from Properties file and the Import from Database jobs). After you activate the job, you must manually add input CIs to the job so that it runs against a particular destination. For details, see "Add the Discovered Shell CI to the Job" on page 393.

Discovery Pattern Parameters

The following parameters are included by default:

- ➤ **csvFile**. The full path to the CSV file on the remote machine. The job uses the Shell CI Type as input to reach this path on the remote machine.
- ➤ **delimiter**. The delimiter used in the CSV file. The comma (,) delimiter is the default but other delimiters are supported. For details, see "Delimiters" on page 379.
- ➤ mappingFile. For details of the mapping file, see "The External Source Mapping Files" on page 386.
- ➤ rowToStartIndex. For details on setting the row at which DDM starts collecting data, see "CSV Files with Column Titles in First Row" on page 375.

For details on overriding a pattern parameter, see "Override Pattern Parameters" on page 418.

Delimiters, Quotes, and Escaping Characters

Delimiters

The delimiter divides values in the same row of a CSV file. Supported delimiters are:

- ➤ **Single symbol**. Any symbol can be used as a delimiter, for example, the pipe sign (|), the letter **O**. Delimiters are case sensitive.
- ➤ **ASCII code.** If an integer number is used as the value for a delimiter parameter, this value is treated as ASCII code, and the related symbol is used as the delimiter. For example, **9** is a valid delimiter because **9** is the ASCII code for the horizontal tab.
- ➤ Known character sequence. A sequence of characters can be used to represent special characters. For example, \t represents the horizontal tab.

Quotation Marks

You can use double or single quotes in values, that is, all values residing between the two quotes are treated as a single value.

➤ If a delimiter symbol is used in a value, the value must be surrounded with quotation marks. For example, the following row includes a comma inside a value, so the value must be quoted:

```
Usain, "25 Hope Road, Kingston", Jamaica
```

➤ If a quote character is used in a value, the character must be escaped by inserting a backslash before it:

```
Usain \"Lightening\" Bolt, "August 21, 1986"
```

This row contains two values:

- 1) Usain "Lightening" Bolt
- 2) August 21, 1986.

Escaping Symbols

The following symbols must always be quoted or escaped:

- ➤ Backslash
- ➤ Single quote
- ➤ Double quote
- ➤ Delimiter, that is, the delimiter used in the same CSV file.

The Import from Properties File Job

This job imports information from a properties file, maps the information to one CI, and imports that CI into the UCMDB.

This section includes the following topics:

- ➤ "Job Details" on page 380
- ➤ "Discovery Pattern Parameters" on page 381
- ➤ "Keys and Values" on page 381
- ➤ "Comments in Properties Files" on page 381

Job Details

The job details are as follows:



This job has no Trigger TQLs associated with it.

Discovery Pattern Parameters

The following parameters are included by default:

- ➤ mappingFile. For details of the mapping file, see "The External Source Mapping Files" on page 386.
- ➤ propertyFile. The full path to the properties file located on a remote machine. The Input CI runs the Shell discovery that is used to access this file on the remote machine.

For details on overriding a pattern parameter, see "Override Pattern Parameters" on page 418.

Keys and Values

Keys cannot contain the equals symbol (=).

Each value must be set out in a single line. Use **backslash+n** (\n) to specify a new line. Values can contain anything, including \n for a new line, quotes, tabs, and so on.

Comments in Properties Files

To create a commented line in a properties file, add the pound sign (#) as the first character in a line. The job ignores commented lines.

The Import from Database Job

This job uses a database table or database query as the source of the information, maps the information to CIs, and imports the CIs into the UCMDB.

This section includes the following topics:

- ➤ "Job Details" on page 382
- ➤ "Discovery Pattern Parameters" on page 382
- ➤ "Tables and Queries" on page 383
- ➤ "Database, Schema, and Table Names" on page 383

- ➤ "Importing Data with a SQL Query" on page 384
- ➤ "Column Types" on page 384

Job Details

The job details are as follows:



This job has no Trigger TQLs associated with it.

Discovery Pattern Parameters

The following parameters are included by default:

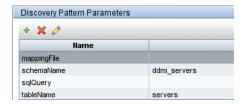
- ➤ mappingFile. The name of the table column or query result column is used as a value for the column attribute in the mapping file. For details of the mapping file, see "The External Source Mapping Files" on page 386.
- > schemaName. The name of the database schema.
- ➤ **sqlQuery**. If a SQL query is specified, mapping is performed against its result. This parameter is ignored if **tableName** is defined.
- ➤ tableName. If a table name is specified, mapping is performed against the table's columns.

For details on overriding a pattern parameter, see "Override Pattern Parameters" on page 418.

Tables and Queries

The following use cases are supported by the Import from Database job (a single SQL query is performed):

➤ Import data using the schema name and table name parameters:



The SQL query is generated from these parameters.

➤ Import data specifying an arbitrary SQL query as the source of the data:



The SQL query is generated from the defined query. For more details, see "Importing Data with a SQL Query" on page 384.

Database, Schema, and Table Names

SQL naming conventions suggest a usage of a <database.schema.table> syntax for the fully qualified name of a table. Note, however, that each vendor treats the specification in a different way. DDM uses the following notation:

- ➤ The **schemaName** parameter specifies the name of a database.
- ➤ The **tableName** parameter specifies the name of a table.

➤ A schema name cannot be specified in a parameter but can be included in a SQL query.

For Oracle, the SQL query is:

```
SELECT * FROM <schemaName.tableName>
```

For Microsoft SQL Server, the SQL query is:

```
SELECT * FROM dbo.tableName
```

Note: The default dbo schema is used for Microsoft SQL Server.

Importing Data with a SQL Query

You can use arbitrarily-complex SQL query expressions, for example, joins, sub-selects and other options, as long as the query is valid and complies with the database usage. Currently, you must use a fully-qualified table name in the query according to the specific database.

Column Types

Types enable you to specify, in the mapping file, the type of column that exists in the external source. For example, a database includes information about column types, and the value of this type needs to be included in the CI's attributes. This is done by adding a **type** element to the **map** element (in mapping_[your mapping file name].xml):

```
<column type="int"></column>
```

Supported type attributes are:

- ➤ string
- ➤ Boolean
- ➤ date

- ➤ int
- ➤ long
- ➤ double
- ➤ float
- **➤** timestamp

Note:

- ➤ You use the **type** attribute for database mapping only.
- ➤ If the column element does not include a type attribute, the element is mapped as a string.

Example of adding a type attribute

A database column has an integer type and can be either 0 or 1. This integer must be mapped to a Boolean attribute of a CIT in the UCMDB. Use the binaryIntToBoolean converter, as follows:

```
<map>
    <attribute>cluster_is_active</attribute>
    <column type="int">cluster_is_active</column>
    <converter module="import_converters">binaryIntToBoolean</converter>
</map>
```

type="int". This attribute specifies that the value of cluster_is_active should be retrieved as an integer, and that the value passed to the converter method should be an integer.

If the cluster_is_active attribute of the CIT is of type integer, the converter is not needed here, and the mapping file should say:

```
<map>
    <attribute>cluster_is_active</attribute>
    <column type="int">cluster_is_active</column>
</map>
```

\lambda The External Source Mapping Files

The data in the external source is mapped to a CI's attributes in the UCMDB by means of a mapping file. The mapping files are located in the **Discovery Resources pane** > **External_source_import package** > **Configuration Files** folder:

- ➤ mapping_template.xml. A template that serves as a source for creating the mapping file.
- ➤ mapping_schema.xsd. The XML schema used to validate the XML mapping file. The XML mapping file must be compliant with this schema.
- ➤ mapping_doc.xml. A file that contains Help on creating a mapping file, including all valid elements.

The mapping file describes the mapping only and does not include information about how data should be obtained. In this way, you can use one mapping file across different jobs.

All the pattern files in the External_source_import package include a mappingFile parameter, for example:

<parameter name="mappingFile" type="string" description="Mapping file located in "Configuration Files" folder of this package" />

name="mappingFile". The value of this parameter is the mapping XML file. The mapping file is always located on the server and is downloaded to the Probe machine upon job execution.

Converters

Converters enable you to specify the way data should be converted between the external source and a CI's attributes.

A CSV file contains records of type string. However, some of the record values need to be handled as numbers. This is done by adding a **converter** element to the **map** element (in [your mapping file name].xml):

<converter module="import converters"></converter>

The **import_converters.py** file contains a set of the most commonly needed converters and types:

- ➤ toString
- ➤ stringToInt
- ➤ stringToLong
- ➤ stringToFloat
- ➤ stringToBoolean
- ➤ stringToDate
- ➤ stringToDouble
- ➤ skipSpaces
- ➤ binaryIntToBoolean
- ➤ stringToBytesArray
- ➤ stringToZippedBytesArray

To access the file: **Discovery Resources pane > External_source_import package > Scripts**.

Example of a Converter

A CSV file contains the following row:

Usain, 21, Male

This row must be mapped to the **Person** CIT that includes name (Usain), age (21), and gender (Male) attributes. The age attribute should be of type **integer**. Therefore, the string in the CSV file must be converted to an integer in the CIT to make it compliant with the CIT attribute type, before the Person CIs can retrieve the age values.

This is done by adding a **converter** element to the **map** element:

```
<map>
    <attribute>age</attribute>
    <column>2</column>
    <converter module="import_converters">stringToInt</converter>
</map>
```

module="import_converters". This attribute specifies from which module the converter is to be retrieved. A module is a Jython script file that contains a set of converter methods, in this case, import_converters.py.

stringToInt. The name of the converter. In the import_converters.py file, the method is written as follows:

```
def stringToInt(value):
if value is not None:
    return int(value.strip())
else:
    return 0
```

Custom Converters

You can write your own custom converters: Add a new method to the import_converters.py file or create your own script and add a set of converter methods to it. Call the method with the name of the script, for example:

```
<converter module="your_converter_script">[your_converter_method]
</converter>
```

🦒 Import CSV Data from an External Source – Scenario

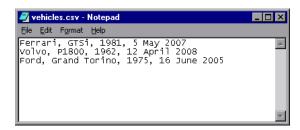
The UCMDB administrator must model a vehicle catalog that is stored in a CSV file.

This task includes the following steps:

- ➤ "Prerequisites" on page 389
- ➤ "Create a CIT" on page 390
- ➤ "Create a Mapping File" on page 390
- ➤ "Activate the Import from CSV File Job" on page 392
- ➤ "Add the Discovered Shell CI to the Job" on page 393
- ➤ "Result" on page 393

1 Prerequisites

The admin opens the CSV file and analyzes the data:



The file includes the name, model, year of manufacture, and the date when the car was purchased, that is, there are four columns of data:

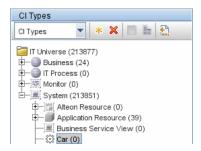
1	Name	string
2	Model	string
3	Year of manufacture	integer
4	Date of purchase	date

There are three rows to the file, which means that the admin expects three CIs to be created in UCMDB.

2 Create a CIT

The admin creates a CIT.

a The admin creates a CIT named **Car** to hold the attributes that are to be mapped to the data in the CSV file (name, model, and so on):



For details, see "Create a CI Type" in Model Management.

b During the creation of the CIT, the admin adds these attributes as follows:



For details, see "Attributes Page" in Model Management.

3 Create a Mapping File

The admin uses the template (mapping_template.xml) to create a mapping file that makes the information available to the Import_CSV pattern. The mapping file is located in the following folder: Manage Discovery Resources > Discovery Resources > External_source_import > Configuration Files.

a For each attribute, the admin adds a <map> marker:

```
<?xml version="1.0" encoding="UTF-8"?>
<mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xsi:noNamespaceSchemaLocation=".\mapping_schema.xsd"
parserClassName="com.hp.ucmdb.discovery.library.communication.downloader.cfg"
files.CiMappingConfigFile">
<ci type="car">
    <map>
        <attribute>name</attribute>
        <column>1</column>
    </map>
    <map>
        <attribute>model</attribute>
        <column>2</column>
    </map>
    <map>
        <attribute>year_of_manufacture</attribute>
        <column>3</column>
    </map>
    <map>
        <attribute>date_of_purchase</attribute>
        <column>4</column>
    </map>
</ci>
</mappings>
```

b The admin then adds information about the attribute type:

```
<?xml version="1.0" encoding="UTF-8"?>
<mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</p>
xsi:noNamespaceSchemaLocation=".\mapping schema.xsd"
parserClassName="com.hp.ucmdb.discovery.library.communication.downloader.cfg"
files.CiMappingConfigFile">
<ci type="">
<map>
    <attribute>name</attribute>
    <column>1</column>
</map>
<map>
    <attribute>model</attribute>
    <column>2</column>
</map>
<map>
    <attribute>year of manufacture</attribute>
    <column>3</column>
    <converter module="import_converters">stringToInteger</converter>
</map>
<map>
    <attribute>date of purchase</attribute>
    <column>4</column>
    <converter module="import_converters">stringToDate</converter>
</map>
</mappings>
```

All conversions between the values in the CSV file and the CI attributes are done by a converter. Several converter types are included in the package by default. For details, see "Converters" on page 386.

4 Activate the Import from CSV File Job

This job uses the Shell Trigger CIT to discover the CSV file on a remote machine. The Input CI Type is Shell and the Discovered CIs are the IT Universe.

The admin activates the following job: Advanced Mode > Discovery Modules > Discovery Tools > Import from CSV file.

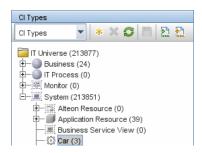
For details on activating jobs, see "Discovery Modules Pane" on page 119.

5 Add the Discovered Shell CI to the Job

After activation, the admin locates the Shell CI (of the machine where the cars.csv file is located) and adds it to the job. For details, see "Choose CIs to Add Dialog Box" on page 96.

6 Result

The admin accesses the CIT Manager and searches for instances of the **Car** CIT. UCMDB finds the three instances of the CIT:



Troubleshooting and Limitations

This section includes the following topics:

- ➤ "DDM Adds Extra CI When Importing from CSV File" on page 393
- ➤ "Timeout Issues When Importing from CSV and Properties Files" on page 394

DDM Adds Extra CI When Importing from CSV File

Problem. When CIs imported from a CSV file are displayed in the Statistics Results pane, one more CI than expected is included in the results. This is because the first row of the CSV file contains column headings that are considered as CIs.

Solution. For details on defining from which row DDM should read the CSV file, see "CSV Files with Column Titles in First Row" on page 375.

Timeout Issues When Importing from CSV and Properties Files

Problem. When importing large CSV or properties files on the network, there may be timeout issues.

Solution. Make sure the files are not large.

10

Content Development and Pattern-Writing

This chapter describes the approaches, methodologies, and practices of developing new Discovery and Dependency Mapping (DDM) content (known as pattern-writing) for HP Business Availability Center.

This chapter includes:

Concepts

- ➤ Introducing Content Development and Pattern-Writing on page 396
- ➤ Associating Business Value with Discovery Development on page 397
- ➤ DDM Patterns and Related Components on page 398
- ➤ The DDM Development Cycle on page 399
- ➤ DDM and Integration on page 403
- ➤ Research Stage on page 404
- ➤ Separating Patterns on page 409
- ➤ HP Discovery and Dependency Mapping API Reference on page 410

Tasks

- ➤ Implement a Pattern on page 411
- ➤ Step 1: Create a Discovery and Dependency Mapping Pattern on page 411
- ➤ Step 2: Assign a Job to the Pattern on page 420
- ➤ Step 3: Create Jython Code on page 422
- ➤ Record DDM Code on page 436
- ➤ Configure Eclipse to Run Jython Scripts in Debug Mode on page 438

Reference

- ➤ Discovery and Dependency Mapping Code on page 447
- ➤ Jython Libraries and Utilities on page 449
- ➤ Using External Java JAR Files Within Jython on page 453
- ➤ Job and Pattern XML Formats on page 453

Introducing Content Development and Pattern-Writing

Prior to beginning actual planning for development of new content, it is important for you to understand the processes and interactions commonly associated with this development.

The following sections can help you understand what you need to know and do, to successfully manage and execute a discovery development project.

This chapter:

- ➤ Assumes a working knowledge of HP Business Availability Center and some basic familiarity with the elements of the DDM system. It is meant to assist you in the learning process and does not provide a complete guide.
- ➤ Covers the stages of planning, research, and implementation of new discovery content for HP Business Availability Center using DDM, along with guidelines and considerations that need to be taken into account.
- ➤ Provides information on the key APIs of the Discovery and Dependency Mapping Framework. For full documentation on the available APIs, see the HP Discovery and Dependency Mapping API Reference. (Other non-formal APIs exist but even though used on out of the box patterns, they may be subject to change.)

🖧 Associating Business Value with Discovery Development

The use case for developing new discovery content should be driven by a business case and plan to produce business value. That is, the goal of mapping system components to CIs and adding them to the CMDB is to provide business value.

The content may not always be used for application mapping, although this is a common intermediate step for many use cases. Regardless of the end usage of the content, your plan should answer these questions of this approach:

- ➤ Who is the consumer? How should the consumer act on the information provided by the CIs (and the relationships between them)? What is the business context in which the CIs and relationships are to be viewed? Is the consumer of these CIs a person or a product or both?
- ➤ Once the perfect combination of CIs and relationships exists in the CMDB, how do I plan on using them to produce business value?
- ➤ What should the perfect mapping look like?
 - ➤ What term would most meaningfully describe the relationships between each CI?
 - ➤ What types of CIs would be most important to include?
 - ➤ What is the end usage and end user of the map?
- ➤ What would be the perfect report layout?

Once the business justification is established, the next step is to embody the business value in a document. This means picturing the perfect map using a drawing tool and understanding the impact and dependencies between CIs, reports, how changes are tracked, what change is important, monitoring, compliance, and additional business value as required by the use cases.

This drawing (or model) is referred as the **blueprint**.

For example, if it is critical for the application to know when a certain configuration file has changed, the file should be mapped and linked to the appropriate CI (to which it relates) in the drawn map.

Work with an SME (Subject Matter Expert) of the area, who is the end user of the developed content. This expert should point out the critical entities (CIs with attributes and relationships) that must exist in the CMDB to provide business value.

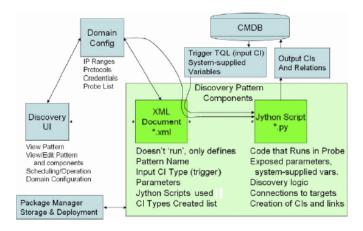
One method could be to provide a questionnaire to the application owner (also the SME in this case). The owner should be able to specify the above goals and blueprint. The owner must at least provide a current architecture of the application.

You should map critical data only and no unnecessary data: you can always enhance DDM later. The goal should be to set up a limited discovery that works and provides value. Mapping large quantities of data gives more impressive maps but can be confusing and time consuming to develop.

Once the model and business value is clear, continue to the next stage. This stage can be revisited as more concrete information is provided from the next stages.

DDM Patterns and Related Components

The following diagram shows a pattern's components and the components they interact with to execute discovery. The components in green are the actual patterns, and the components in blue are components that interact with patterns.



Note that the minimum notion of a pattern is two files: an XML document and a Jython script. The DDM Framework, including input CIs, credentials, and user-supplied libraries, is exposed to the pattern at run time. Both discovery pattern components are administered through the DDM application. They are stored operationally in the CMDB itself; although the external package remains, it is not referred to for operation. The Package Manager enables preservation of the new discovery content capability.

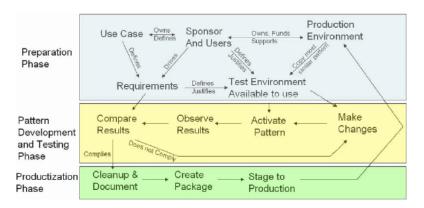
Input CIs to the pattern are provided by a TQL, and are exposed to the pattern script in system-supplied variables. Pattern parameters are also supplied as destination data, so you can configure the pattern's operation according to a pattern's specific function.

The DDM application is used to create and test new patterns. You use the job, resource, and domain configuration pages during pattern-writing.

Patterns are stored and transported as packages. The Package Manager application and the JMX console are used to create packages from newly created patterns, and to deploy patterns on new systems.

The DDM Development Cycle

The following illustration shows a flowchart for pattern-writing. Most of the time is spent in the middle section which is the iterative loop of development and testing.



Each phase of pattern development builds on the last one:

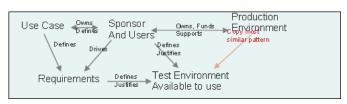
- ➤ The Research and Preparation Phase phase encompasses the driving business needs and use cases, and also accounts for securing the necessary facilities to develop and test the pattern.
- ➤ The Pattern Development and Testing phase is a highly iterative process. As the pattern begins to take shape, you begin testing against the final use cases, make changes, test again, and repeat this process until the pattern complies with the requirements.
- ➤ The Pattern Packaging and Productization phase accounts for the last phase of development. As a best practice, a final pass should be made to clean up debugging remnants, documents and comments, to look at security considerations, and so on, before moving on to packaging. You should always have at least a readme document to explain the inner workings of the pattern. Someone (maybe even you) may need to look at this pattern in the future and will be aided greatly by even the most limited documentation.

Once you are satisfied with the way the pattern looks and works, you are ready to package it. Using either the UCMDB Package Manager or manual exporting of the components, create a DDM package *.zip file. As a best practice, you should deploy and test this package on another UCMDB system before releasing it to production, to ensure that all the components are accounted for and successfully packaged. For details on packaging, see "Create a Custom Package" in *Model Management*.

The following sections expand on each of the phases showing the most critical steps and best practices.

Research and Preparation Phase





- 1 When planning to modify an existing pattern, the first technical step is to make a backup of that pattern and ensure you can return it to its pristine state. If you plan to create a new pattern, copy the most similar pattern and save it under an appropriate name. For details, see "<Pattern files>" on page 218.
- **2** Research how the pattern should collect data.
 - ➤ Use External tools/protocols to obtain the data
 - ➤ Develop how the pattern should create CIs based on the data
 - ➤ You now know what a similar pattern should look like
- **3** Determine most similar pattern based on:
 - ➤ Same CIs created
 - ➤ Same Protocols used (SNMP)
 - ➤ Same kind of targets (by OS type, versions, and so on)
- **4** Copy entire package.
- **5** Unzip into work space and rename the pattern (XML) and Jython (.py) files.



Pattern Development and Testing

Startup and Preparation of Copy

- ➤ Modify XML parts of the pattern: Name (id) in line 1, Created CI Types, and Called Jython script name.
- ➤ Get the copy running with identical results to the original pattern.
- ➤ Comment out most of the code, especially the critical result-producing code.

Development and Testing

- ➤ Use other sample code to develop changes
- ➤ Test pattern by running it
- ➤ Use a dedicated view to validate complex results, search to validate simple results

Pattern Packaging and Productization

Cleanup and Document

- ➤ Remove debugging
- ➤ Comment all functions and add some opening comments in the main section
- ➤ Create sample TQL and view for the user to test

Create Package

- ➤ Export patterns, TQL, and so on with the Package Manager. For details, see "Package Manager" in *Model Management*.
- ➤ Check any dependencies your package has on other packages, for example, if the CIs created by those packages are input CIs to your pattern.
- ➤ Use Package Manager to create a package zip. For details, see "Package Manager" in *Model Management*.
- ➤ Test deployment by removing parts of the new content and redeploying, or deploying on another test system.

& DDM and Integration

DDM discovery patterns are capable of integration with other products. Consider the following definitions:

- ➤ DDM collects specific content from many targets.
- ➤ Integration collects multiple types of content from one system.

Note that these definitions do not distinguish between the methods of collection. Neither does DDM. The process of developing a new pattern is the same process for developing new integration. You do the same research, make the same choices for new vs. existing patterns, write the patterns the same way, and so on. Only a few things change:

- ➤ The final pattern's scheduling. Integration patterns may run more frequently than discovery, but it depends on the use cases.
- ➤ Input CIs:
 - ➤ Integration: non-CI trigger to run with no input: a file name or source is passed through the pattern parameter.
 - ➤ Discovery: uses regular, UCMDB CIs for input.

For integration projects, you should almost always reuse an existing pattern. The direction of the integration (from HP Business Availability Center to another product, or from another product to HP Business Availability Center) may affect your approach to development. There are field packages available for you to copy for your own uses, using proven techniques.

From HP Business Availability Center to another project:

- ➤ Create a TQL that produces the CIs and relations to be exported.
- ➤ Use a generic wrapper pattern to execute the TQL and write the results to an XML file for the external product to read.

Note: For examples of field packages, contact HP Software Support.

Chapter 10 • Content Development and Pattern-Writing

To integrate another product to HP Business Availability Center: Depending on how the other product exposes its data, the integration pattern acts differently:

Integration Type	Reference Example to Be Reused
Access the product's database directly	HP ED
Read in a csv or xml file produced by an export	HP ServiceCenter
Access a product's API	BMC Atrium/Remedy

Research Stage

The prerequisite of this stage is a **blueprint** of the CIs and relationships needed to be discovered by DDM, which should include the attributes that are to be discovered. For details, see "Introducing Content Development and Pattern-Writing" on page 396.

This section includes the following topics:

- ➤ "Modifying an Existing Pattern" on page 405
- ➤ "Writing a New Pattern" on page 405
- ➤ "Model Research" on page 406
- ➤ "Technology Research" on page 406
- ➤ "Guidelines for Choosing Ways to Access Data" on page 407
- ➤ "Summary" on page 408

Modifying an Existing Pattern

You modify an existing pattern when an out-of-the-box or field DDM pattern exists, but:

- ➤ it does not discover specific attributes that are needed
- ➤ a specific type of target (OS) is not being discovered or is being incorrectly discovered
- ➤ a specific relationship is not being discovered or created

If an existing pattern does some, but not all, of the job, your first approach should be to evaluate the existing patterns and verify if one of them almost does what is needed; if it does, you can modify the existing pattern.

You should also evaluate if an existing field pattern is available. Field patterns are discovery patterns that are available but are not out-of-the-box. Contact HP Software Support to receive the current list of field patterns.

Writing a New Pattern

A new pattern needs to be developed:

- ➤ When it is faster to write a pattern than to insert the information manually into the CMDB (generally, from about 50 to 100 CIs and relationships) or it is not a one-time effort.
- ➤ When the need justifies the effort.
- ➤ If out of the box or field patterns are not available.
- ➤ If the results can be reused.
- ➤ When the target environment or its data is available (you cannot discover what you cannot see).

Model Research

- ➤ Browse the CMDB class model (CI Type Manager) and match the entities and relations from your **blueprint** to existing CITs. It is highly recommended to adhere to the current model to avoid possible complications during version upgrade. If you need to extend the model, you should create new CITs since an upgrade may overwrite out of the box CITs.
- ➤ If some entities, relations, or attributes are lacking from the current model, you should create them. It is preferable to create a package with these CITs (which will also later hold all the discovery, views, and other artifacts relating to this package) since you need to be able to deploy these CITs on each installation of HP Business Availability Center.

Technology Research

Once you have verified that the CMDB hold the relevant CIs, the next stage is to decide how to retrieve this data from the relevant systems.

Retrieving data usually involves using a protocol to access a management part of the application, actual data of the application, or configuration files or databases that are related to the application. Any data source that can provide information on a system is valuable. Technology research requires both extensive knowledge of the system in question and sometimes creativity.

For home-grown applications, it may be helpful to provide a questionnaire form to the application owner. In this form the owner should list all the areas in the application that can provide information needed for the blueprint and business values. This information should include (but does not have to be limited to) management databases, configuration files, log files, management interfaces, administration programs, Web services, messages or events sent, and so on.

For off-the-shelf products, you should focus on documentation, forums, or support of the product. Look for administration guides, plug-ins and integrations guides, management guides, and so on. If data is still missing from the management interfaces, read about the configuration files of the application, registry entries, log files, NT event logs, and any artifacts of the application that control its correct operation.

Guidelines for Choosing Ways to Access Data

Relevance: Select sources or a combination of sources that provide the most data. If a single source supplies most information whereas the rest of the information is scattered or hard to access, try to assess the value of the remaining information by comparison with the effort or risk of getting it. Sometimes you may decide to reduce the blueprint if the value or cost does not warranty the invested effort.

Reuse: If HP Business Availability Center already includes a specific connection protocol support it is a good reason to use it. It means the DDM Framework is able to supply a ready made client and configuration for the connection. Otherwise, you may need to invest in infrastructure development. You can view the currently supported HP Business Availability Center connection protocols: **Discovery** > **Setup Discovery Probe** > **Domains and Probes pane**. For details, see "Domains and Probes Pane" on page 173.

You can add new protocols by adding new CIs to the model. For details, contact HP Software Support.

Note: To access Windows Registry data, you can use either WMI or NTCmd.

Security: Access to information usually requires credentials (user name, password), which are entered in the CMDB and are kept secure throughout the product. If possible, and if adding security does not conflict with other principles you have set, choose the least sensitive credential or protocol that still answers access needs. For example, if information is available both through JMX (standard administration interface, limited) and Telnet, it is preferable to use JMX since it inherently provides limited access and (usually) no access to the underlying platform.

Comfort: Some management interfaces may include more advanced features. For example, it might be easier to issues queries (SQL, WMI) than to navigate information trees or build regular expressions for parsing.

Developer Audience: The people who will eventually develop DDM may have an inclination towards a certain technology. This can also be considered if two technologies provide almost the same information at an equal cost in other factors.

Summary

The outcome of this stage is a document describing the access methods and the relevant information that can be extracted from each method. The document should also contain a mapping from each source to each relevant blueprint data.

Each access method should be marked according to the above instructions. Finally you should now have a plan of which sources to discover and what information to extract from each source into the blueprint model (which should by now have been mapped to the corresponding UCMDB model).

Separating Patterns

Technically, an entire discovery could be defined in a single pattern. But good design demands that a complex system be separated into simpler, more manageable components.

The following are guidelines and best practices for dividing the DDM process:

- ➤ Discovery should be done in stages. Each stage should be represented by a pattern that should map an area or tier of the system. Patterns should rely on the previous stage or tier to be discovered, to continue discovery of the system. For example, Pattern A is triggered by an application server TQL result and maps the application server tier. As part of this mapping, a JDBC connection component is mapped. Pattern B registers a JDBC connection component as a trigger TQL and uses the results of pattern A to access the database tier (for example, through the JDBC URL attribute) and maps the database tier.
- ➤ The two-phase connect paradigm: Most systems require credentials to access their data. This means that a user/password combination needs to be tried against these systems. The DDM administrator supplies credentials information in a secure way to the system and can give several, prioritized login credentials. This is referred to as the Protocol Dictionary. If the system is not accessible (for whatever reason) there is no point in performing further discovery. If the connection is successful, there needs to be a way to indicate which credential set was successfully used, for future discovery access.

These two phases lead to a separation of the two patterns in the following cases:

- ➤ Connection Pattern: This is a pattern that accepts an initial trigger and looks for the existence of a remote agent on that trigger. It does so by trying all entries in the Protocol Dictionary which match this agent's type. If successful, this pattern provides as its result a remote agent CI (SNMP, WMI, and so on), which also points to the correct entry in the Protocol Dictionary for future connections. This agent CI is then part of a trigger for the content pattern.
- ➤ Content Pattern: This pattern's precondition is the successful connection of the previous pattern (preconditions specified by the TQLs). These types of patterns no longer need to look through all of the Protocol Dictionary since they have a way to obtain the correct credentials from the remote agent CI and use them to log in to the discovered system.
- ➤ Different scheduling considerations can also influence discovery division. For example, a system may only be queried during off hours, so even though it would make sense to join the pattern to the same pattern discovering another system, the different schedules mean that you need to create two patterns.
- ➤ Discovery of different management interfaces or technologies to discover the same system should be placed in separate patterns. This is so that you can activate the access method appropriate for each system or organization. For example, some organizations have WMI access to machines but do not have SNMP agents installed on them.

A HP Discovery and Dependency Mapping API Reference

For full documentation on the available APIs, see *HP Discovery and Dependency Mapping API Reference*. These files are located in the following folder:

\\<HP Business Availability Center Gateway Server root directory>\
AppServer\webapps\site.war\amdocs\eng\doc_lib\
Discovery_and_Dependency_Mapping\DDM_JavaDoc\index.html

🏲 Implement a Pattern

A DDM task has the aim of accessing remote (or local) systems, modeling extracted data as CIs, and saving the CIs to the CMDB. The task consists of the following steps:

1 DDM pattern.

You configure a pattern file that holds the context, parameters, and result types for DDM by selecting the scripts that are to be part of the pattern. For details, see the following section.

2 DDM job.

You configure a job with scheduling information and a trigger TQL. For details, see "Step 2: Assign a Job to the Pattern" on page 420.

3 DDM code.

You can edit the Jython or Java code that is contained in the pattern files and that refers to the DDM Framework. For details, see "Step 3: Create Jython Code" on page 422.

To write new DDM content, you create each of the above components, each one of which is automatically bound to the component in the previous step. For example, once you create a job and select the relevant pattern, the pattern file binds to the job.

Step 1: Create a Discovery and Dependency Mapping Pattern

A DDM pattern can be considered as the definition of a function. This function defines an input definition, runs logic on the input, defines the output, and provides a result.

Each DDM pattern specifies input and output: Both input and output are Trigger CIs that are specifically defined in the pattern. The DDM pattern extracts data from the input Trigger CI and passes this data as parameters to the DDM code. (Data from related CIs is sometimes passed to the code too. For details, see "Related CIs Window" on page 152.) A pattern's DDM code is generic, apart from these specific input Trigger CI parameters that are passed to the code.

For details on input components, see "Trigger CITs, Trigger CIs, Input TQLs, and Trigger TQLs" on page 50.

This section includes the following topics:

- ➤ "Define Pattern Input (Trigger CIT and Input TQL)" on page 412
- ➤ "Define Pattern Output" on page 417
- ➤ "Override Pattern Parameters" on page 418

P Define Pattern Input (Trigger CIT and Input TQL)

You use the Trigger CIT and Input Topology Query Language (TQL) components to define specific CIs as pattern input:

- ➤ The Trigger CIT defines which CIT is used as the input for the pattern. For example, for a pattern that is going to discover IPs, the input CIT is Network.
- ➤ The Input TQL is a regular, editable TQL that defines the query against the CMDB. The Input TQL defines additional constraints on the CIT (for example, if the task requires a hostID or application ip attribute), and can define more CI data, if needed by the pattern.
 - If the pattern requires additional information from the CIs that are related to the Trigger CI, you can add additional nodes to the input TQL. For details, see "Example of Input TQL Definition" on page 413 and "Add Nodes and Relationships to a TQL Query" in Model Management.
- ➤ The Trigger CI data contains all the required information on the Trigger CI as well as information from the other nodes in the Input TQL, if they are defined. DDM uses variables to retrieve data from the CIs. When the task is downloaded to the Probe, the Trigger CI data variables are replaced with actual values that exist on the attributes for real CI instances.

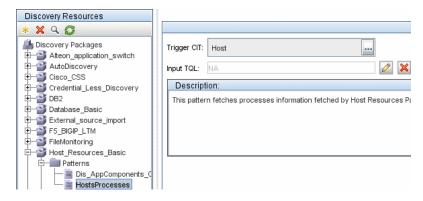
Example of Trigger CIT Definition

In this example, a Trigger CIT defines that IP CIs are permitted in the pattern.

1 Access Discovery > Manage Discovery Resources > Pattern Signature. Select the HostProcesses pattern (Discovery Packages > Host_Resources_Basic > Patterns > HostProcesses).

- **2** Locate the Trigger CIT box. For details, see "Triggered CI Data Pane" on page 239.
- **3** Click the button to open the Choose Discovered Class dialog box. For details, see "Choose Discovered Class Dialog Box" on page 210.
- 4 Select the CIT.

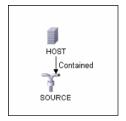
In this example, the IP CI (Host) is permitted in the pattern:



Example of Input TQL Definition

In this example, the Input TQL defines that the IP CI (configured in the previous example as the Trigger CIT) must be connected to a Host CI.

- 1 Access Discovery > Manage Discovery Resources > Pattern Signature. Locate the Input TQL box. Click the Edit button to open the Input TQL Editor. For details, see "Input TQL Editor Window" on page 220.
- **2** In the Input TQL Editor, name the Trigger CI node **SOURCE**: right-click the node and choose **Node Properties**. In the **Element Name** box, change the name to **SOURCE**.
- **3** Add a host and a Contains relationship to the IP CI. For details on working with the Input TQL Editor, see "Input TQL Editor Window" on page 220.

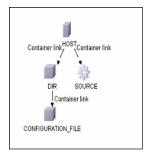


The **IP** CI is connected to a **HOST** CI. The input TQL consists of two nodes, **HOST** and **IP**, with a link between them. The **IP** CI is named **SOURCE**.

Example of Adding Variables to the Input TQL

In this example, you add DIRECTORY and CONFIGURATION_FILE variables to the Input TQL created in the previous example. These variables help to define what must be discovered, in this case, to find the configuration files residing on the hosts that are linked to the IPs you need to discover.

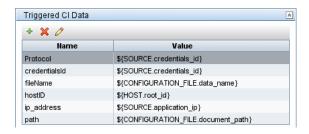
- **1** Display the Input TQL created in the previous example.
- **2** Access **Discovery > Manage Discovery Resources > Pattern Signature**. Locate the Triggered CI Data pane. For details, see "Triggered CI Data Pane" on page 239.
- **3** Add variables to the Input TQL. For details, see "Value" on page 306.



Example of Replacing Variables with Actual Data

In this example, variables replace the IP CI data variables with actual values that exist on real IP CI instances in your system.

The Triggered CI data for the IP CI includes a fileName variable. This variable enables the replacement of the CONFIGURATION_FILE node in the Input TQL with the actual values of the configuration file located on a host:

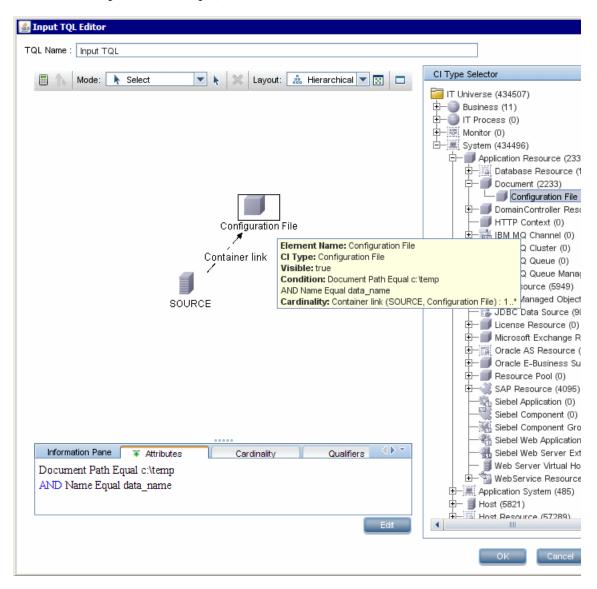


The Trigger CI data is uploaded to the Probe with all variables replaced by actual values. The pattern script includes a command to use the DDM Framework to retrieve the actual values of the defined variables:

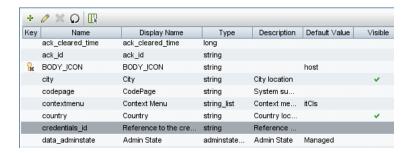
Framework.getTriggerCIData ('ip_address')

Note:

➤ The fileName and path variables use the data_name and document_path attributes from the Configuration File node (defined in the Input TQL – see previous example).



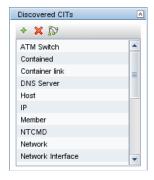
➤ The Protocol, credentialsId, and ip_address variables use the root_class, credentials id, and application ip attributes:



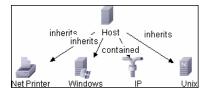


Define Pattern Output

The output of the pattern is new CIs and links between them. That is, the output is a list of discovered CITs (**Discovery > Manage Discovery Resources > Pattern Signature tab**):



You can also view the CITs as a topology map, that is, the components and the way in which they are linked together (click the **View Discovered CITs as Map** button):



The discovered CIs are returned by the DDM code (that is, the Jython script) in the format of UCMDB's ObjectStateHolderVector. For details, see "Results Generation by the Jython Script" on page 428.

Example of Pattern Output

In this example, you define which CITs are to be part of the IP CI output.

- 1 Access Discovery > Manage Discovery Resources.
- 2 In the Discovery Resources pane, select Network > Pattern > NSLOOKUP_on_Probe.
- **3** In the Pattern Signature tab, locate the Discovered CITs pane.
- **4** The CITs that are to be part of the pattern output are listed. Add CITs to, or remove from, the list. For details, see "Discovered CITs Pane" on page 236.



Override Pattern Parameters

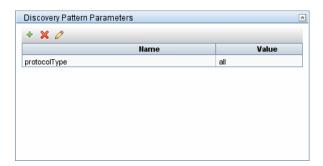
To configure a pattern for more than one job, you can override pattern parameters. For example, the pattern SQL NET Dis Connection is used by both the MSSQL Connection by SQL and the Oracle Connection by SQL jobs.

Example of Overriding a Pattern Parameter

This example illustrates overriding a pattern parameter so that one pattern can be used to discover both Microsoft SQL Server and Oracle databases.

- 1 Access Discovery > Manage Discovery Resources.
- **2** In the Discovery Resources pane, select **Database Basic > Pattern > SQL NET Dis Connection.**

3 In the Pattern Signature tab, locate the **Discovery Pattern Parameters** pane. The protocolType parameter has a value of **all**:



- **4** Right-click the **SQL_NET_Dis_Connection** pattern and choose **Go to Discovery Job** > **MSSQL Connection by SQL**.
- **5** Display the Properties tab. Locate the Parameters pane:



The all value is overwritten with the MicrosoftSQLServer value.

Note: The Oracle Connection by SQL job includes the same parameter but the value is overwritten with an oracle value.

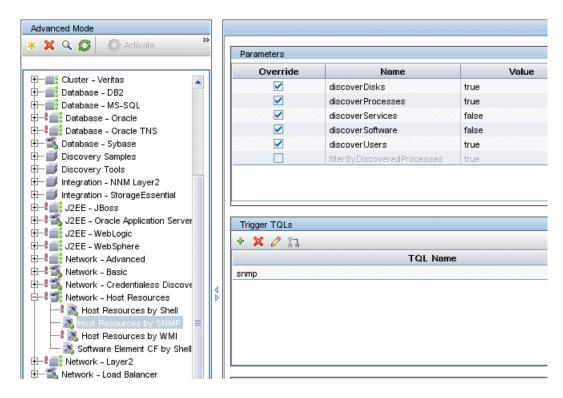
For details on adding, deleting, or editing parameters, see "Discovery Pattern Parameters Pane" on page 236.

DDM begins looking for Microsoft SQL Server instances according to this parameter.

🦒 Step 2: Assign a Job to the Pattern

Each pattern has one or more associated jobs that define the execution policy. Jobs enable scheduling the same pattern differently over different set of Triggered CIs and also enable supplying different parameters for each set.

The jobs appear in the Discovery Modules tree, and this is the entity that the user activates.



Trigger TQL

Each job is associated with Trigger TQLs. These Trigger TQLs publish results that are used as Input Trigger CIs for the pattern of this job.

A Trigger TQL can add constraints to an Input TQL. For example, if an input TQL's results are IPs connected to SNMP, a trigger TQL's results can be IPs connected to SNMP within the range 195.0.0.0-195.0.0.10.

Note: A trigger TQL must refer to the same objects that the input TQL refers to. For example, if an input TQL queries for IPs running SNMP, you cannot define a trigger TQL (for the same job) to query for IPs connected to a host, because some of the IPs may not be connected to an SNMP object, as required by the input TQL.

Scheduling

The scheduling information for the Probe specifies when to run the code on Trigger CIs. If the **Invoke on new triggered CIs Immediately** check box is selected, the code also runs once on each Trigger CI when it reaches the Probe, regardless of future schedule settings.



For each schedule occurrence for each job, the Probe runs the DDM code against all Trigger CIs accumulated for that job. For details, see "Schedule Modules to Run" on page 53 and "Discovery Scheduler Dialog Box" on page 124.

Parameters

When configuring a job you can override the pattern parameters. For details, see "Override Pattern Parameters" on page 418.

Step 3: Create Jython Code

"DDM code is mostly written in Jython. Jython is an implementation of the high-level, dynamic, object-oriented language Python written in 100% pure Java, and seamlessly integrated with the Java platform. It thus enables you to run Python on any Java platform." (Quoted from the Jython Web site http://www.jython.org/Project/index.html.)

The following section describes the actual writing of Jython code inside the DDM Framework. This section specifically addresses those contact points between the Jython script and the Framework that it calls, and also describes the Jython libraries and utilities that should be used whenever possible.

Note:

- ➤ Scripts written for DDM should be compatible with Jython version 2.1.
- ➤ For full documentation on the available APIs, see the *HP Discovery and Dependency Mapping API Reference*.

This section includes the following topics:

- ➤ "Execution of the Code" on page 423
- ➤ "Modifying Out of the Box Scripts" on page 423
- ➤ "Structure of the Jython File" on page 425
- ➤ "Results Generation by the Jython Script" on page 428
- ➤ "The Framework Instance" on page 430
- ➤ "Finding the Correct Credentials (for Connection Patterns)" on page 434
- ➤ "Handling Exceptions from Java" on page 436
- ➤ "Jython Libraries and Utilities" on page 449
- ➤ "Using External Java JAR Files Within Jython" on page 453



P Execution of the Code

After a job is activated, a task with all the required information is downloaded to the Probe.

The Probe starts running the DDM code using the information specified in the task.

The Jython code flow starts running from a main entry in the script, executes code to discover CIs, and provides results of a vector of discovered CIs.



Modifying Out of the Box Scripts

When making out of the box script modifications, make only minimal changes to the script and place any necessary methods in an external script. You can track changes more efficiently and, when moving to a newer HP Business Availability Center version, your code is not overwritten.

For example, the following single line of code in an out of the box script calls a method that calculates a Web server name in an application-specific way:

serverName = iplanet_cspecific.PlugInProcessing(serverName, transportHN, mam utils)

Chapter 10 • Content Development and Pattern-Writing

The more complex logic that decides how to calculate this name is contained in an external script:

implement customer specific processing for 'servername' attribute of httpplugin # def PlugInProcessing(servername, transportHN, mam_utils_handle): # support application-specific HTTP plug-in naming if servername == "appsrv instance": # servername is supposed to match up with the j2ee server name, however some groups do strange things with their # iPlanet plug-in files. this is the best work-around we could find. this join can't be done with IP address:port # because multiple apps on a web server share the same IP:port for multiple websphere applications logger.debug('httpcontext webapplicationserver attribute has been changed from [' + servername + '] to [' + transportHN[:5] + '] to facilitate websphere enrichment') servername = transportHN[:5] return servername

Save the external script in the External Resources folder. For details, see "Discovery Resources Pane" on page 215. If you add this script to a package, you can use this script for other jobs, too. For details on working with Package Manager, see "Package Manager" in *Model Management*.

During upgrade, the change you make to the single line of code is overwritten by the new version of the out of the box script, so you will need to replace the line. However, the external script is not overwritten.



Structure of the Jython File

The Jython file is composed of three parts in a specific order:

- ➤ 1. Imports
- ➤ 2. Functions definitions (optional)
- ➤ 3. Main Function DiscoveryMain

The following is an example of a Jython script:

imports section

from appilog.common.system.types import ObjectStateHolder from appilog.common.system.types.vectors import ObjectStateHolderVector

Function definition

def foo:

do something

Main Function

def DiscoveryMain(Framework): OSHVResult = ObjectStateHolderVector()

Write implementation to return new result CIs here...

return OSHVResult

Imports

Jython classes are spread across hierarchical namespaces. In version 7.0 or later, unlike in previous versions, there are no implicit imports, and so every class you use must be imported explicitly. (This change was made for performance reasons and to enable an easier understanding of the Jython script by not hiding necessary details.)

➤ To import a Jython script:

import logger

➤ To import a Java class:

from appilog.collectors.clients import ClientsConsts

Main Function - DiscoveryMain

Each Jython runable script file contains a main function: DiscoveryMain.

The DiscoveryMain function is the main entry into the script; it is the first function that runs. The main function may call other functions that are defined in the scripts:

def DiscoveryMain(Framework):

The Framework argument must be specified in the main function definition. This argument is used by the main function to retrieve information that is required to run the scripts (such as information on the Trigger CI and parameters) and can also be used to report on errors that occur during the script run.

You can create a Jython script without any main method. Such scripts are used as library scripts that are called from other scripts.

Functions Definition

Each script can contain additional functions that are called from the main code. Each such function can call another function, which either exists in the current script or in another script (use the import statement). Note that to use another script, you must add it to the Scripts section of the package:



Example of a Function Calling Another Function

In the following example, the main code calls the doQueryOSUsers(..) method which calls an internal method doOSUserOSH(..):

```
def doOSUserOSH(name):
sw obj = ObjectStateHolder('winosuser')
sw obj.setAttribute('data name', name)
# return the object
return sw_obj
def doQueryOSUsers(client, OSHVResult):
hostObj = modeling.createHostOSH(client.getIpAddress())
data name mib = '1.3.6.1.4.1.77.1.2.25.1.1,1.3.6.1.4.1.77.1.2.25.1.2,string'
resultSet = client.executeQuery(data name mib)
while resultSet.next():
    UserName = resultSet.getString(2)
    OSUserOSH = doOSUserOSH(UserName)
    OSUserOSH.setContainer( hostObj)
    OSHVResult.add(OSUserOSH)
def DiscoveryMain(Framework):
OSHVResult = ObjectStateHolderVector()
    client =
Framework.getClientFactory(ClientsConsts.SNMP PROTOCOL NAME).createClient()
    Framework.reportError('Connection failed')
else:
    doQueryOSUsers(client, OSHVResult)
    client.close()
return OSHVResult
```

If this script is a global library that is relevant to many patterns, you can add it to the list of scripts in the jythonGlobalLibs.xml configuration file, instead of adding it to each pattern (Discovery > Manage Discovery Resources > Discovery Packages > AutoDiscovery > Configuration Files).



Results Generation by the Jython Script

Each Jython script runs on a specific Trigger CI, and ends with results that are returned by the return value of the DiscoveryMain function.

The script result is actually a group of CIs and links that are to be inserted or updated in the CMDB. The script returns this group of CIs and links in the format of ObjectStateHolderVector.

The ObjectStateHolder class is a way to represent an object or link defined in the CMDB. The ObjectStateHolder object contains the CIT name and a list of attributes and their values. The ObjectStateHolderVector is a vector of ObjectStateHolder instances.

The ObjectStateHolder Syntax

This section explains how to build the DDM results into a CMDB model.

Example of Setting Attributes on the Cls

The ObjectStateHolder class describes the DDM result graph. Each CI and link (relationship) is placed inside an instance of the ObjectStateHolder class as in the following Jython code sample:

```
# siebel application server
1 appServerOSH = ObjectStateHolder('siebelappserver')
2 appServerOSH.setStringAttribute('data name', sblsvrName)
3 appServerOSH.setStringAttribute ('application_ip', ip)
4 appServerOSH.setContainer(appServerHostOSH)
```

- ➤ Line 1 creates a CI of type siebelappserver.
- ➤ Line 2 creates an attribute called **data name** with a value of **sblsvrName** which is a Jython variable set with the value discovered for the server name.
- ➤ Line 3 sets a non-key attribute that is updated in the CMDB.
- ➤ Line 4 is the building of containment (the result is a graph). It specifies that this application server is contained inside a host (another ObjectStateHolder class in the scope).

Note: Each CI being reported by the Jython script must include values for all the key attributes of the CI's CI Type.

Example of Relationships (Links)

The following link example explains how the graph is represented:

```
1 linkOSH = ObjectStateHolder('route')
2 linkOSH.setAttribute('link_end1', gatewayOSH)
3 linkOSH.setAttribute('link_end2', appServerOSH)
```

- ➤ Line 1 creates the link (that is also of the ObjectStateHolder class. The only difference is that route is a link CI Type).
- ➤ Lines 2 and 3 specify the nodes at the end of each link. This is done using the end1 and end2 attributes of the link which must be specified (because they are the minimal key attributes of each link). The attribute values are ObjectStateHolder instances. For details on End 1 and End 2, see "Link" on page 211.

Important: A link is directional. You should verify that End 1 and End 2 nodes correspond to valid CITs at each end. If the nodes are not valid, the result object fails DDM validation and is not reported correctly. For details, see "CI Type Relationships" in *Model Management*.

Example of Vector (Gathering Cls)

After creating objects with attributes, and links with objects at their ends, you must now group them together. You do this by adding them to an ObjectStateHolderVector instance, as follows:

```
oshvMyResult = ObjectStateHolderVector()
oshvMyResult.add(appServerOSH)
oshvMyResult.add(linkOSH)
```

For details on reporting this composite result to the Framework so it can be sent to the CMDB server, see the sendObjects method.

Once the DDM result graph is assembled in an ObjectStateHolderVector instance, it must be returned to the DDM Framework to be inserted into the CMDB. This is done by returning the ObjectStateHolderVector instance as the result of the DiscoveryMain() function.

Note: For details on creating **OSH** for common CITs, see **modeling.py** in "Jython Libraries and Utilities" on page 449.



The Framework Instance

The Framework instance is the only argument that is supplied in the main function in the Jython script. This is an interface that can be used to retrieve information required to run the script (for example, information on trigger CIs and pattern parameters), and is also used to report on errors that occur during the script run. For details, see "HP Discovery and Dependency Mapping API Reference" on page 410.

This section describes the most important Framework usages:

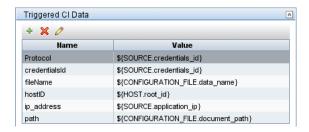
- ➤ "Framework.getTriggerCIData(String attributeName)" on page 430
- ➤ "Framework.createClient(credentialsId, props)" on page 431
- ➤ "Framework.getParameter (String parameterName)" on page 433
- ➤ "Framework.reportError(String message) and Framework.reportWarning(String message)" on page 433

Framework.getTriggerCIData(String attributeName)

This API provides the intermediate step between the Trigger CI data defined in the pattern and the script.

Example of Retrieving Credential Information

You request the following Trigger CI data information:



To retrieve the credential information from the task, use this API:

credId = Framework.getTriggerCIData('credentialsId')

Framework.createClient(credentialsId, props)

You make a connection to a remote machine by creating a client object and executing commands on that client. To create a client, retrieve the ClientFactory class. The getClientFactory() method receives the type of the requested client protocol. The protocol constants are defined in the ClientsConsts class. For details on credentials and supported protocols, see "Domain Credential References" on page 177.

Example of Creating a Client Instance for the Credentials ID

To create a Client instance for the credentials ID:

```
properties = Properties()
codePage = Framework.getCodePage()
properties.put( BaseAgent.ENCODING, codePage)
client = Framework.createClient(credentailsID, properties)
```

You can now use the Client instance to connect to the relevant machine or application.

Example of Creating a WMI Client and Running a WMI Query

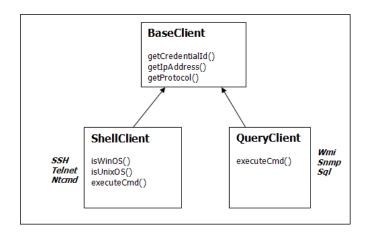
To create a WMI client and run a WMI query using the client:

```
wmiClient = Framework.createClient(credential)
resultSet = wmiClient. executeQuery("SELECT TotalPhysicalMemory
FROMWin32_LogicalMemoryConfiguration")
```

Note: To make the createClient() API work, add the following parameter to the Trigger CI data parameters: **credentialsId = \${SOURCE.credentials_id}** in the Triggered CI Data pane. Or you can manually add the credentials ID when calling the function:

wmiClient = clientFactory().createClient(credentials_id).

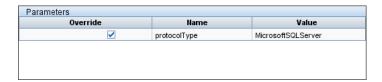
The following diagram illustrates the hierarchy of the clients, with their commonly-supported APIs:



For details on the clients and their supported APIs, see BaseClient, ShellClient, and QueryClient in the *HP Discovery and Dependency Mapping API Reference*.

Framework.getParameter (String parameterName)

In addition to retrieving information on the Trigger CI, you often need to retrieve a pattern parameter value. For example:



Example of Retrieving the Value of the protocolType Parameter

To retrieve the value of the protocolType parameter from the Jython script, use the following API:

protocolType = Framework.getParameterValue('protocolType')

Framework.reportError(String message) and Framework.reportWarning(String message)

Some errors (for example, connection failure, hardware problems, timeouts) can occur during a script run. When such errors are detected, Framework can report on the problem. The message that is reported reaches the server and is displayed for the user.

Example of a Report Error and Message

The following example illustrates the use of the reportError(<Error Msg>) API:

```
try:
    client = Framework.getClientFactory(ClientsConsts.SNMP_PROTOCOL_NAME)
    createClient()
except:
    strException = str(sys.exc_info()[1]).strip()
Framework. reportError ('Connection failed: %s' % strException)
```

You can use either one of the APIs—Framework.reportError(String message), Framework.reportWarning(String message)—to report on a problem. The difference between the two APIs is that when reporting an error, the Probe saves a communication log file with the entire session's parameters to the file system. In this way you are able to track the session and better understand the error.

Finding the Correct Credentials (for Connection Patterns)

A pattern trying to connect to a remote system needs to try all possible credentials. One of the parameters needed when creating a client (through ClientFactory) is the credentials ID. The connection script gains access to possible credential sets and tries them one by one using the clientFactory.getAvailableProtocols() method. When one credential set succeeds, the pattern reports a CI connection object on the host of this trigger CI (with the credentials ID that matches the IP) to the CMDB. Subsequent patterns can use this connection object CI directly to connect to the credential set (that is, the patterns do not have to try all possible credentials again).

The following example shows how to obtain all entries of the SNMP protocol. Note that here the IP is obtained from the Trigger CI data (# Get the Trigger CI data values).

The connection script requests all possible protocol credentials (# Go over all the protocol credentials) and tries them in a loop until one succeeds (resultVector). For details, see the **two-phase connect paradigm** entry in "Separating Patterns" on page 409.

```
import logger
from appilog.collectors.clients import ClientsConsts
from appilog.common.system.types.vectors import ObjectStateHolderVector
     def mainFunction(Framework):
resultVector = ObjectStateHolderVector()
    # Get the Trigger CI data values
    ip address = Framework.getDestinationAttribute('ip address')
    ip domain = Framework.getDestinationAttribute('ip domain')
    # Create the client factory for SNMP
     clientFactory = framework.getClientFactory(ClientsConsts.SNMP PROTOCOL NAME)
    protocols = clientFactory.getAvailableProtocols(ip_address, ip_domain)
     connected = 0
    # Go over all the protocol credentials
    for credentials id in protocols:
         client = None
         try:
             # try to connect to the snmp agent
             client = clientFactory.createClient(credentials_id)
             // Query the agent
             # connection succeed
             connected = 1
         except:
             if client != None:
                 client.close()
     if (not connected):
         logger.debug('Failed to connect using all credentials')
    else:
         // return the results as OSHV
         return resultVector
```

🏲 Handling Exceptions from Java

Some Java classes throw an exception upon failure. It is recommended to catch the exception and handle it, otherwise it causes the pattern to terminate unexpectedly.

When catching a known exception, in most cases you should print its stack trace to the log and issue a proper message to the UI, for example:

```
try:
     client = Framework.getClientFactory().createClient()
except Exception, msg:
     Framework.reportError('Connection failed')
     logger.debugException('Exception while connecting: %s' % (msg))
     return
```

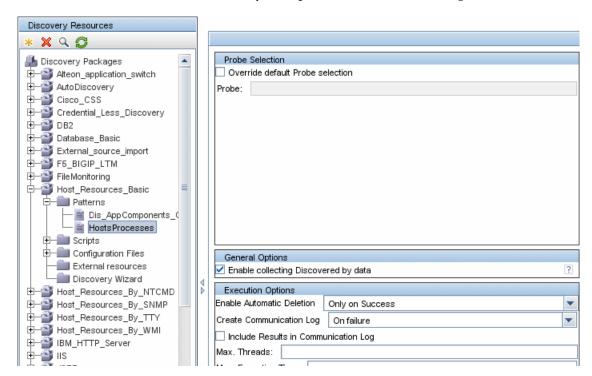
If the exception is not fatal and the script can continue, you should omit the call for the reportError() method and enable the script to continue.

Record DDM Code

It can be very useful to record an entire execution, including all parameters, for example, when debugging and testing code. This task describes how to record an entire execution with all relevant variables. Furthermore, you can view extra debug information that is usually not printed to log files even at the debug level.

To record DDM code:

1 Access Discovery > Run Discovery. Right-click the job whose run must be logged and select **Edit pattern** to open the Manage Discovery Resources application.



2 Locate the **Execution Options** pane in the Pattern Management tab:

3 Change the **Create communication logs** box to **Always**. For details on setting logging options, see "Execution Options Pane" on page 230.

The following example is the XML log file that is created when the Host Connection by Shell job is run and the Create communication logs box is set to Always or On Failure:

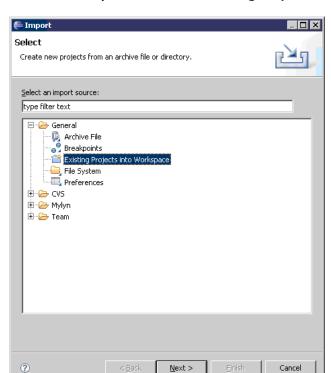
The following example shows the message and stacktrace parameters:

🦒 Configure Eclipse to Run Jython Scripts in Debug Mode

This task explains how to configure Eclipse so that you can run your Jython scripts in debug mode, thus enabling better visibility to job threads, trigger CIs, and results.

To configure Eclipse:

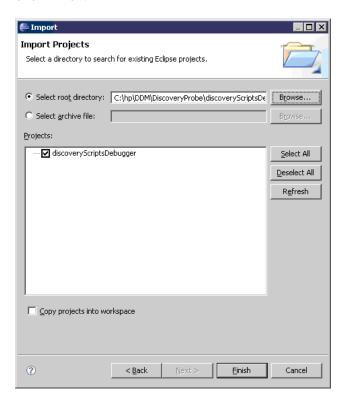
- **1** Prerequisites:
 - ➤ Install Java 5.0 SE on your computer (if it is not already installed). The files are available on the Java Developers Web site http://www.java.sun.com.
 - ➤ Install the latest Eclipse version on your computer. The application is available at www.eclipse.org.
 - ➤ Install PyDev and PyDev Extensions, available at: http://pydev.sourceforge.net/ (PyDev) and http://www.fabioz.com/pydev/ (PyDev Extensions)
 - ➤ The Discovery Probe must be installed on this computer.



2 Select File > Import > General > Existing Projects into Workspace:

3 Click Next. Click Browse to select the following directory:
C:\hp\DDM\DiscoveryProbe\discoveryScriptsDebugger:

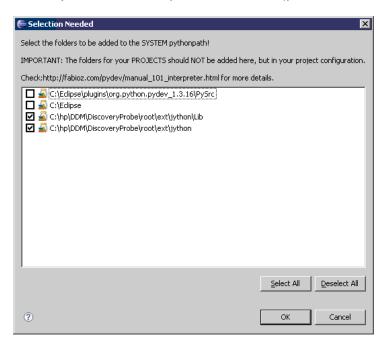
4 Verify that **discoveryScriptsDebugger** is selected in the Projects pane and click **Finish**:



A discoveryScriptsDebugger project is added to the Eclipse workspace.

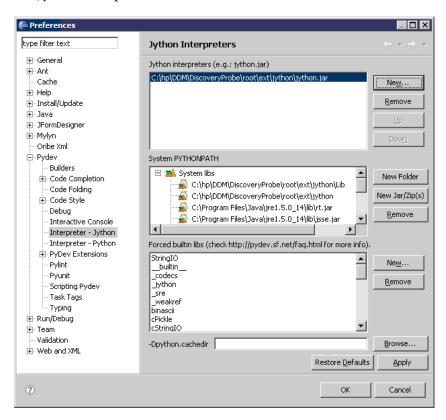
5 Select **Window** > **Preferences** to open the **Preferences** dialog box. Select **Pydev** > **Interpreter** - **Jython** > **New**.

- **6** Select the following directories:
 - C:\hp\DDM\DiscoveryProbe\root\ext\jython\
 - ➤ C:\hp\DDM\DiscoveryProbe\root\ext\jython\lib



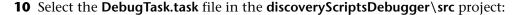
Click **OK**.

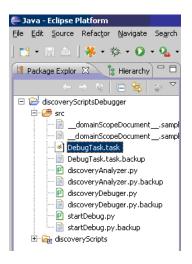
7 Return to the **Preferences** dialog box. Verify that the jython.jar file appears in the Jython interpreters list:



Click OK.

- **8** Select Window > Open Perspective > Other to open the Open Perspective dialog box. Select PyDev.
- **9** You can configure the job to be debugged either in a text editor or by using a graphic utility. For details on the text editor configuration, go to the next step. For details on the graphic utility configuration, skip to step 13 on page 445.



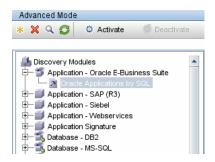


To edit this file in Eclipse, right-click **DebugTask.task** and select **Open With** > **Text Editor**.

The DebugTask.task file is a template you can use to debug your Jython scripts. It is located in the following directory:

C:\hp\DDM\DiscoveryProbe\discoveryScriptsDebugger\src.

11 Change the jobid value to the name of the job to be debugged. For example, delete Range IPs by ICMP and enter Oracle Applications by SQL. You can copy the name from the job name in the Run Discovery page:



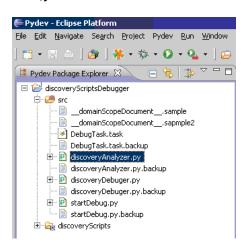
12 Add as many destinationData attributes as necessary (do not change the destinationList className). For example, to debug the **Oracle Applications by SQL** script, enter the following values:

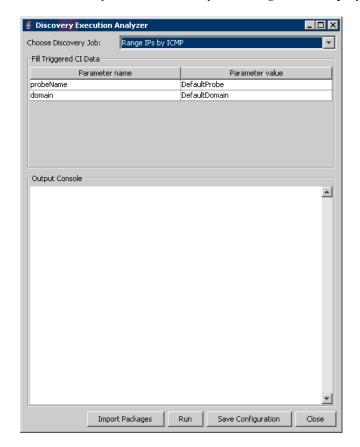
You can verify the exact destinationData attributes for each job by consulting the following log file: C:\hp\DDM\DiscoveryProbe\root\logs\probeGW-taskResults.log.

Note: Disregard UPDATE_SERVER_DATA jobs in the log file.

Skip to step 14 on page 446.

13 To use the graphic utility, select the **discoveryAnalyzer.py** file in the **discoveryScriptsDebugger\src** project. Right-click the file and choose **Run** as > **Jython run**.





The Discovery Execution Analyzer dialog box is displayed:

Choose the Discovery Job, fill in the parameter values and save the configuration.

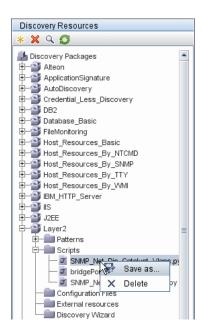
- **14** Add a breakpoint in the Jython script to be debugged.
- **15** In the discoveryScriptsDebugger project, right-click startDebug.py and choose Debug As > Jython Run.

Discovery and Dependency Mapping Code

The actual implementation of connecting to the remote system, querying its data, and mapping it as CMDB data is performed by the DDM code. For example, the code contains the logic for connecting to a database and extracting data from it. In this case, the code expects to receive a JDBC URL, a user name, a password, a port, and so on. These parameters are specific for each instance of the database that answers the TQL query. You define these variables in the pattern (in the Trigger CI data) and when the job runs, these specific details are passed to the code for execution.

The pattern can refer to this code by a Java class name or a Jython script name. In this section we discuss writing DDM code as Jython scripts.

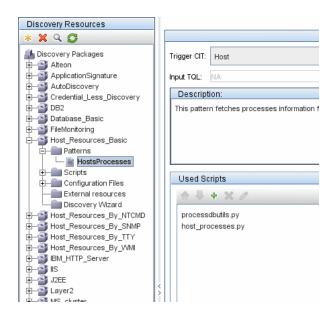
A pattern can contain a list of scripts to be used when running DDM. When creating a new pattern, you usually create a new script and assign it to the pattern. A new script includes basic templates, but you can use one of the other scripts as a template by right-clicking it and selecting **Save as**:



For details on writing new Jython scripts, see "Step 3: Create Jython Code" on page 422. You add scripts through the Manage Discovery Resources window:



The list of scripts are run one after the other, in the order in which they are defined in the pattern:



Note: A script must be specified even though it is being used solely as a library by another script. In this case, the library script must be defined before the script using it. In this example, the processedbutils.py script is a library used by the last host_processes.py script. Libraries are distinguished from regular runable scripts by the lack of the DiscoveryMain() function.

💘 Jython Libraries and Utilities

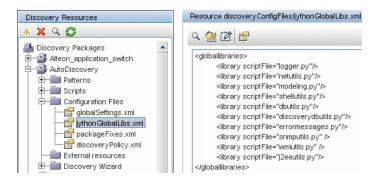
Several utility scripts are used widely in DDM patterns. These scripts are part of the AutoDiscovery package and are located under: C:\hp\DDM \DiscoveryProbe\root\lib\collectors\probeManager\discoveryScripts with the other scripts that are downloaded to the Probe.

Note: The discoveryScript folder is created dynamically when the Probe begins working.

To use one of the utility scripts, add the following import line to the import section of the script:

import <script name>

The AutoDiscovery Python library contains Jython utility scripts. These library scripts are considered DDM's external library. They are defined in the jythonGlobalLibs.xml file (located in the **Configuration Files** folder).



Each script that appears in the jythonGlobalLibs.xml file is loaded by default at Probe startup, so there is no need to use them explicitly in the pattern definition.

This section includes the following topics:

- ➤ "logger.py" on page 450
- ➤ "modeling.py" on page 452
- ➤ "netutils.py" on page 452
- ➤ "shellutils.py" on page 452

logger.py

The **logger.py** script contains log utilities and helper functions for error reporting. You can call its debug, info, and error APIs to write to the log files. Log messages are recorded in

Messages are entered in the log file according to the debug level defined for the PATTERNS_DEBUG appender in the C:\hp\DDM\DiscoveryProbe\root \lib\collectors\probeManager\probeMgrLog4j.properties file. (By default, the level is DEBUG.) For details, see "Severity Levels" on page 54.

```
################
                                        PATTERNS DEBUG loa
log4j.category.PATTERNS DEBUG=DEBUG, PATTERNS DEBUG
log4j.appender.PATTERNS DEBUG=org.apache.log4j.RollingFileAppender
log4j.appender.PATTERNS DEBUG.File=C:/hp/DDM/DiscoveryProbe/root/logs/probe
Mgr-patternsDebug.log
log4j.appender.PATTERNS_DEBUG.Append=true
log4j.appender.PATTERNS DEBUG.MaxFileSize=15MB
log4j.appender.PATTERNS_DEBUG.Threshold=DEBUG
log4j.appender.PATTERNS DEBUG.MaxBackupIndex=10
log4j.appender.PATTERNS DEBUG.layout=org.apache.log4j.PatternLayout
log4i.appender.PATTERNS DEBUG.layout.ConversionPattern=<%d> [%-5p] [%t] -
%m%n
log4j.appender.PATTERNS_DEBUG.encoding=UTF-8
```

The info and error messages also appear in the Command Prompt console.

There are two sets of APIs:

- ➤ logger.<debug/info/warn/error>
- ➤ logger.<debugException/infoException/warnException/errorException>

The first set issues the concatenation of all its string arguments at the appropriate log level and the second set issues the concatenation as well as issuing the stack trace of the most recently-thrown exception, to provide more information, for example:

```
logger.debug('found the result')
logger.errorException('Error in discovery')
```

modeling.py

The **modeling.py** script contains APIs for creating hosts, IPs, process CIs, and so on. These APIs enable the creation of common objects and make the code more readable. For example:

```
ipOSH= modeling.createlpOSH(ip)
host = modeling.createHostOSH(ip_address)
member1 = modeling.createLinkOSH('member', ipOSH, networkOSH)
```

netutils.py

The **netutils.py** library is used to retrieve network and TCP information, such as retrieving operating system names, checking if a MAC address is valid, checking if an IP address is valid, and so on. For example:

```
dnsName = netutils.getHostName(ip, ip)
isValidIp = netutils.isValidIp(ip_address)
address = netutils.getHostAddress(hostName)
```

shellutils.py

The **shellutils.py** library provides an API for executing shell commands and retrieving the end status of an executed command, and enables running multiple commands based on that end status. The library is initialized with a Shell Client, and uses the client to run commands and retrieve results. For example:

```
ttyClient = clientFactory.createClient(Props)
clientShUtils = shellutils.ShellUtils(ttyClient)
if (clientShUtils.isWinOs()):
    logger.debug ('discovering Windows..')
```

🔍 Using External Java JAR Files Within Jython

When developing new Jython scripts, external Java Libraries (JAR files) or third party executable files are sometimes needed as either Java utility archives, connection archives such as JDBC Driver JAR files, or executable files (for example, nmap.exe is used for credential-less discovery).

These resources should be bundled in the package under the External Resources folder. Any resource put in this folder is automatically sent to any Probe that connects to your HP Universal CMDB server.

In addition, when discovery is launched, any JAR file resource is loaded into the Jython's classpath, making all the classes within it available for import and use.

1 Job and Pattern XML Formats

Jobs and patterns are saved in the CMDB in an XML format. The job name appears in the job's XML file and is referred to by the **id** attribute. Each job has a corresponding pattern which is referred to by the **patternId** attribute.

Example of Job XML

A job name is **CPUs by TTY**. Its corresponding pattern is **TTY_HR_CPU**:

Example of Pattern XML

The pattern name is **TTY_HR_CPU**. The pattern input is defined by the **<inputClass>** tag. The pattern output is defined by the **<discoveredClasses>** tag.

```
<pattern id="TTY_HR_CPU" description="Discover CPU on Unix boxes."</pre>
schemaVersion="7.0">
<discoveredClasses>
    <discoveredClass>container f</discoveredClass>
    <discoveredClass>cpu</discoveredClass>
</discoveredClasses>
<parameters />
<taskInfo className="appilog.collectors.services.dynamic.core.DynamicService">
    <destinationInfo className="appilog.collectors.tasks.BaseDestinationData">
        <destinationData
name="ip address">${SOURCE.application ip}</destinationData>
        <destinationData
name="Protocol">${SOURCE.root class}</destinationData>
        <destinationData
name="credentialsId">${SOURCE.credentials id}</destinationData>
        <destinationData
name="language">${SOURCE.language:NA}</destinationData>
        <destinationData
name="codepage">${SOURCE.codepage:NA}</destinationData>
    </destinationInfo>
    <params</pre>
className="appilog.collectors.services.dynamic.core.DynamicServiceParams">
        <script>TTY HR CPU Lib.py</script>
        <script>TTY HR CPU.py</script>
    </params>
</taskInfo>
<inputClass>shell</inputClass>
</pattern>
```

11

The HP Discovery and Dependency Mapping Web Service API

This chapter explains how third-party or custom tools can use the HP Discovery and Dependency Mapping Web Service to manage Discovery and Dependency Mapping (DDM).

For full documentation on the available operations, see *HP Discovery and Dependency Mapping Schema Reference*. These files are located in the following folder:

\\<HP Business Availability Center Gateway Server root directory>\
AppServer\webapps\site.war\amdocs\eng\doc_lib\
Discovery_and_Dependency_Mapping\DDM_Schema\webframe.html

This chapter includes:

Concepts

- ➤ Conventions on page 456
- ➤ The HP Discovery and Dependency Mapping Web Service on page 456

Tasks

➤ Call the Web Service on page 458

Reference

➤ Discovery and Dependency Mapping Methods on page 458

Conventions

This chapter uses the following conventions:

- ➤ This style, Element, indicates that an item is an entity in the database or an element defined in the schema, including structures passed to or returned by methods. Plain text indicates that the item is being discussed in a general context.
- ➤ DDM elements and method arguments are spelled in the case in which they are specified in the schema. This usually means that a class name or generic reference to an instance of the class is capitalized. An element or argument to a method is not capitalized. For example, a credential is an element of type Credential passed to a method.

The HP Discovery and Dependency Mapping Web Service

The HP Discovery and Dependency Mapping Web Service is an API used to integrate applications with HP Universal CMDB (UCMDB). The API provides methods to:

- ➤ Manage credentials: view, add, update, and remove
- ➤ Manage jobs: view status, activate, and deactivate
- ➤ Manage probe ranges: view. add, and update
- ➤ Manage triggers: Add or remove a trigger CI, and add, remove, or disable a trigger TQL
- ➤ View general data on domains and probes

Users of the HP Discovery and Dependency Mapping Web Service should be familiar with:

- ➤ The SOAP specification
- ➤ An object-oriented programming language such as C++, C# or Java
- ➤ HP Universal CMDB
- ➤ Discovery and Dependency Mapping

This section includes the following topics:

➤ "Permissions" on page 457

Permissions

The administrator provides login credentials for connecting with the Web service. The required credentials depend on whether you are using DDM with a standalone version of HP Universal CMDB or from within HP Business Availability Center.

When permissions are assigned through DDM, the permission levels are View, Update, and Execute. When they are assigned using the HP Business Availability Center, the levels are View and Update, where Update also includes Execution. To view the permissions required for each operation, see each operation's request documentation in the HP Discovery and Dependency Mapping Schema Reference.

To assign permissions:

- ➤ DDM with HP Universal CMDB standalone. Log in using the credentials of a DDM user who has been granted permissions on the discovery resources.
- ➤ DDM embedded in HP Business Availability Center. Log in using the credentials of a Business Availability Center user. The user must have been granted the relevant permissions on the UCMDB Web Service resource in Business Availability Center.

The Call the Web Service

The HP Discovery and Dependency Mapping Web Service enables calling server-side methods using standard SOAP programming techniques. If the statement cannot be parsed or if there is a problem invoking the method, the API methods throw a SoapFault exception. When a SoapFault exception is thrown, the service populates one or more of the error message, error code, and exception message fields. If there is no error, the results of the invocation are returned.

SOAP programmers can access the WSDL at:

http://<server>[:port]/axis2/services/DiscoveryService?wsdl

The port specification is only necessary for non-standard installations. Consult your system administrator for the correct port number.

The URL for calling the service is:

http://<server>[:port]/axis2/services/DiscoveryService

Discovery and Dependency Mapping Methods

This section contains a list of the Web service operations and a brief summary of their use. For full documentation of the request and response for each operation, see *HP Discovery and Dependency Mapping Schema Reference*.

This section includes the following topics:

- ➤ "Managing Discovery Job Methods" on page 459
- ➤ "Managing Trigger Methods" on page 459
- ➤ "Domain and Probe Data Methods" on page 460
- ➤ "Credentials Data Methods" on page 460
- ➤ "Data Refresh Methods" on page 461

Managing Discovery Job Methods

➤ activateJob

Activates the specified job.

➤ deactivateJob

Deactivates the specified job.

➤ dispatchAdHocJob

Dispatches a job on the probe ad-hoc. The job must be active and contain the specified trigger CI.

➤ getDiscoveryJobsNames

Returns the list of job names.

➤ isJobActive

Checks whether the job is active.

Managing Trigger Methods

➤ addTriggerCl

Adds a new trigger CI to the specified job.

➤ addTriggerTQL

Adds a new trigger TQL to the specified job.

➤ disableTriggerTQL

Prevents the TQL from triggering the job, but does not permanently remove it from the list of queries that trigger the job.

➤ removeTriggerCl

Removes the specified CI from the list of CIs that trigger the job.

➤ removeTriggerTQL

Removes the specified TQL from the list of queries that trigger the job.

➤ setTriggerTQLProbesLimit

Restrict the probes in which the TQL is active in the job to the specified list.

Domain and Probe Data Methods

➤ getDomainType

Returns the domain type.

➤ getDomainsNames

Returns the names of the current domains.

➤ getProbelPs

Returns the IP addresses of the specified probe.

➤ getProbesNames

Returns the names of the probes in the specified domain.

➤ getProbeScope

Returns the scope definition of the specified probe.

➤ isProbeConnected

Checks whether the specified probe is connected.

➤ updateProbeScope

Sets the scope of the specified probe, overriding the existing scope.

Credentials Data Methods

➤ addCredentialsEntry

Adds a credentials entry to the specified protocol for the specified domain.

➤ getCredentialsEntriesIDs

Returns the IDs of the credentials defined for the specified protocol.

➤ getCredentialsEntry

Returns the credentials defined for the specified protocol. Encrypted attributes are returned empty.

> removeCredentialsEntry

Removes the specified credentials from the protocol.

➤ updateCredentialsEntry

Sets new values for properties of the specified credentials entry.

Data Refresh Methods

➤ rediscoverCls

Locates the triggers that discovered the specified CI objects and reruns those triggers.

rediscoverCls runs asynchronously. Call **checkDiscoveryProgress** to determine when the rediscovery is complete.

➤ checkDiscoveryProgress

Returns the progress of the most recent **rediscoverCls** call on the specified IDs. The response is a value from 0 to 1. When the response is 1, the **rediscoverCls** call has completed.

➤ rediscoverViewCls

Locates the triggers that created the data to populate the specified view, and reruns those triggers.

rediscoverViewCls runs asynchronously. Call **checkViewDiscoveryProgress** to determine when the rediscovery is complete.

➤ checkViewDiscoveryProgress

Returns the progress of the most recent **rediscoverViewCls** call on the specified view. The response is a value from 0-1. When the response is 1, the **rediscoverCls** call has completed.

Chapter 11 • The HP Discovery and Dependency Mapping Web Service API

Part IV

Discovery and Dependency Mapping Security

12

Hardening Discovery and Dependency Mapping

This chapter provides information on hardening Discovery and Dependency Mapping.

This chapter includes:

Concepts

- ➤ Hardening Discovery and Dependency Mapping Overview on page 466

 Tasks
- ➤ Manage the Domain Scope Document (DSD) on page 470

Hardening Discovery and Dependency Mapping – Overview

Discovery credentials entered in the Set Up Discovery Probes window are saved in an encrypted file termed a domain scope document (DSD). This DSD contains discovery domain data. Each discovery domain entry in the document contains the network scope for the domain's Probes and the credentials the Probes may use when communicating with remote machines.

Note: Security features related to HP Business Availability Center user management—for example, authentication and authorization—are not discussed here.

This section includes the following topics:

- ➤ "Basic Security Assumptions" on page 466
- ➤ "Credentials Encryption Management" on page 467
- ➤ "HTTPS\SSL Configuration" on page 467

Basic Security Assumptions

Note the following security assumptions:

- ➤ You have secured the HP Business Availability Center Gateway Server and Probe file systems for authorized access only.
- ➤ You have secured the Gateway Server JMX console to enable access to HP Business Availability Center system administrators only, preferably only through localhost access.

Credentials Encryption Management

Note the following guidelines for managing credential encryption:

- ➤ The DSD file is encrypted for both transfer and persistency processes.
- ➤ The DSD file is encrypted to the data stores using standard, symmetric encryption.
- ➤ A default, symmetric key is distributed with the HP Business Availability Center installation. As all default keys are identical, you should replace this key with a locally-generated key.
- ➤ The DSD file is exportable and importable in encrypted file form. To import a file you should supply the matching key used for the encryption of the file. Perform the import and export operations through the server's JMX console (the Discovery Manager service). For details, see "Export and Import DSD File in Encrypted Format" on page 472.
- ➤ You can exchange the key while the system is up to keep the consistency of the system without losing any data. This is managed through the server's JMX console.
- ➤ When a key is updated, you can automatically distribute the new key to Probes. This option is easier to deploy but is considered less secure. The new key is encrypted using the old key.

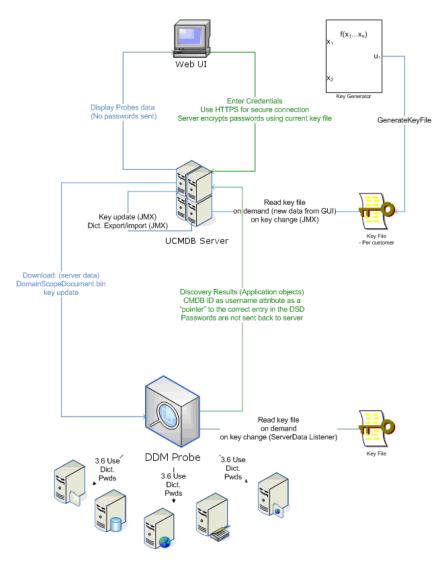
HTTPS\SSL Configuration

You can manage the DSD file using either http or https (for secure mode). You can configure communication between the HP Business Availability Center Gateway Server and the DDM Probe to use https\SSL. This enables better DSD security during transit.

For details on the procedure to configure HP Business Availability Center to work with a SSL, see "Using SSL with the Discovery and Dependency Mapping Probe" in .

Note: As a result of using SSL, other aspects (for example, discovery tasks and gathered results) of the HP Business Availability Center product may become more secure.

The following illustration shows a hardened system.



eals Manage the Domain Scope Document (DSD)

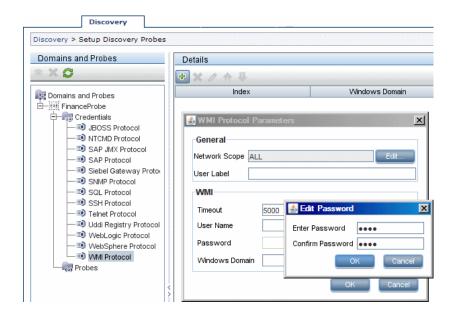
This section explains how to manage the DSD file.

This section includes the following tasks:

- ➤ "View the DSD File (Data Direction: Server to HP Business Availability Center)" on page 470
- ➤ "Update the DSD File (Data Direction: HP Business Availability Center to Server)" on page 471
- ➤ "Change the Location of the DSD File" on page 471
- ➤ "Distribute the Encryption Key" on page 472
- ➤ "Export and Import DSD File in Encrypted Format" on page 472

1 View the DSD File (Data Direction: Server to HP Business Availability Center)

Passwords are not sent from the Server to the application. That is, HP Business Availability Center displays asterisks (*) in the password field, regardless of content:



2 Update the DSD File (Data Direction: HP Business Availability Center to Server)

- ➤ All communication in this direction is not encrypted using a secure key. You must connect to the HP Business Availability Center Gateway Server with a secure communication using https\SSL, or ensure connection through a trusted network.
- ➤ The symmetric key encryption ensures that passwords are not being sent as clear text on the network. This encryption does not use a strong cipher.
- ➤ The password field is limited to 40 characters. The length of the password is not limited in other ways since it is saved only in a file.
- ➤ You can use special characters and non-English characters as passwords.

3 Change the Location of the DSD File

The DSD file (**DomainScopeDocument.bin**) is downloaded to the Probe as part of the downloading mechanism for Probe configuration files.

The symmetric encryption key is located both on the Probe and Server file systems.

By default, the Probe keeps a copy of the DSD file on its file system. You can change this default so that the DSD file is held in the Probe's memory only and is not stored on its file system. In this case, each time the Probe comes up, it must retrieve the DSD file from the Server.

To change this default so that the DSD file is not stored on the Probe's file system:

- **a** Access the DiscoveryProbe.properties file (located in **\<DDM Probe root folder>\root\lib\collectors**).
- **b** Set appilog.collectors.storeDomainScopeDocument to **false** (the default is **true**).

4 Distribute the Encryption Key

You can distribute the encryption key to Probes either offline or online (while the system is running). The key file is a 3DES, 192 bits, binary file.

To distribute the encryption key offline:

- **a** Shut down the HP Business Availability Center Gateway Server and Probes.
- **b** Replace the key file on the Server and Probe file systems.
- **c** Reactivate the HP Business Availability Center Gateway Server and Probes.

The locations of the key are as follows:

- ➤ Gateway Server: <HP Business Availability Center root directory>\lib\server\discovery\key.bin
- ➤ Probe: C:\hp\DDM\DiscoveryProbe\root\lib\collectors\probeManager \binaryData\key.bin.

To update the encryption key online:

- **a** Launch the Web browser and enter the following address: http://localhost.<domain_name>:8080/jmx-console.
- **b** You may have to log in with a user name and password.
- **c** Under MAM, click **service=Discovery manager** to open the JMX MBEAN View page.
- **d** Locate the **changeEncryptionKey** operation and enter the new key file name.
- **e** Click **Invoke** to update the encryption key.

5 Export and Import DSD File in Encrypted Format

You can export and import DSD files in encrypted format. (You would probably import a DSD file during recovery following a system crash or during upgrade.)

To export or import a DSD file:

- **a** Launch the Web browser and enter the following address: http://localhost.<domain_name>:8080/jmx-console.
- **b** You may have to log in with a user name and password.
- **c** Under MAM, click **service=Discovery manager** to open the JMX MBEAN View page.
- **d** Locate the **ExportDomainScopeDocument** or **importDomainScopeDocument** operation and enter the new file name.
- **e** Click **Invoke** to export or import the encryption key.

Chapter 12 • Hardening Discovery and Dependency Mapping

13

Using SSL with the Discovery and Dependency Mapping Probe

This chapter describes how to configure your HP Business Availability Center platform with the Discovery and Dependency Mapping Probe and Staging Data Replicator to support communication using the Secure Sockets Layer (SSL) channel.

For introductory and general information on configuring HP Business Availability Center and its data collectors to support SSL, see "Using SSL in HP Business Availability Center" on page 61.

This chapter includes:

Tasks

➤ Configure SSL from the Discovery Probe to the Gateway Server on page 475

Configure SSL from the Discovery Probe to the Gateway Server

When a session is started between the Discovery Probe and the Gateway Server, the Gateway Server sends the probe a server-side certificate that was issued by a Certification Authority (CA) recognized by the Gateway Server. The Discovery Probe engine should be configured to trust the certificate or the CA that issued it, and to communicate via SSL.

To configure the Discovery Probe to connect to the Gateway Server using SSL:

- **1** Prerequisite: Configure HP Business Availability Center to use SSL.
- **2** Prerequisite: Install the Discovery Probe. During installation, enter the name of the HP Business Availability Center Gateway server to which the Probe must report results.
- **3** If you are working with the Certificate Authority, download the current Certificate Authority certificate to your DDM Probe server. Save it to a file, for example, C:\ca.cer.
- 4 Import this certificate into the DDM Probe JVM:C:\hp\DDM\DiscoveryProbe\jre\bin with the following values:

keytool -import -trustcacerts -alias <your alias> -keystore ..\lib\security\cacerts - file <file path and name>

- **5** Enter the password and click **Yes** to confirm.
- **6** Set the connection parameters in the DDM Probe.
 - ➤ Open the file %discovery root%\root\lib\collectors\
 DiscoveryProbe.properties.
 - ➤ Configure the URL of the HP Business Availability Center server: serverName = <HP Business Availability Center Gateway server domain name>

Note: The SSL connection may fail if an IP address is used instead of domain name.

➤ Configure the port number to use for HTTPS:

Ports used for HTTP/s traffic

#serverPort = 80

serverPortHttps = 443

➤ Set the schema to be used by the Agent to HTTPS:

Can be either HTTP or HTTPS
appilog.agent.probe.protocol = HTTPS

- ➤ Set the name of the HP Business Availability Center server:

 # Name of the Server machine to which this probe reports

 serverName = <server name either of the reverse proxy or the Gateway

 server>
- **7** Restart the Discovery Probe.

Chapter 13 • Using SSL with the Discovery and Dependency Mapping Probe

Index

accessing data guidelines 407 activateJob JMX operations 80 activateJobOnDestination JMX operations 80 Add IP Range dialog box 164 Add New Probe dialog box 167, 168 Add Policy dialog box 166 Advanced Mode window 93 agentless technology 43 API discovery and dependency mapping webservice 455 applicationSignature.xml 200 B	class model
Basic Mode window 94 blueprint 397 Business Availability Center servers 45	Database Wizard 102 Database Port Scanning page Database wizard 101 Database wizard 99
Choose CIs to Add dialog box 96 Choose Discovered CIT dialog box 210 Choose Discovery Jobs dialog box 169 Choose Discovery TQL dialog box 97 Choose Probe to Filter dialog box 98 CIs automatic deletion 199 handling deleted system components 196 manually creating network CI 53 view current status of discovered 252	Custom JDBC Drivers page 102 Database Port Scanning page 101 Define Credentials page 100 Oracle TNSName File Location page 103 Schedule Discovery page 104 Summary page 105 DDM applications 48 architecture 44 code 447

components 45	host resources 315
content 261	IBM DB2 Server 297
development cycle 399	IIS 370
hardening 465	Layer 2 320
integration 403	load balancers 335
introduction 41	Microsoft Cluster Server 292
patterns and related components 398	Microsoft SQL Server 299
user interface 44	network - basic 308
wizards 47	Oracle 301
DDM code	SAP 276
recording 436	Siebel 282
DDM jobs	software elements 198
overview 46	Solaris zones 342
DDM modules	Veritas Cluster Server 294
overview 46	VMware 352
DDM Probe 29, 45	WebLogic 304
automatic CIs deletion 199	WebSphere 305
configuration update 34	Discovery and Dependency Mapping
data validation 34	webservice 455
handling tasks 30	calling 458
installation requirements 28	exceptions 458
installing 36	managing query methods 459
launching 37	mapping methods 458
logs 57	permissions 457
selecting 177	discovery content overview 262
set up 49	Discovery Modules pane 119
setting up 159	Discovery Pattern Parameters pane 236
viewing job information 79	Discovery Pattern Source Editor window 213
DDM server	Discovery Permissions window 123
logs 56	Discovery Probe
Define Credentials page	configuring SSL support for 475
Database wizard 100	using with SSL 475
Infrastructure wizard 130	Discovery Probes pane 170
J2EE wizard 136	Discovery Resources pane 215
Define IP Ranges page	Discovery Scheduler dialog box 124
Infrastructure wizard 128	Discovery Status pane
Dependency Map tab 106	problem management 72
deployment	Discovery Tools 303
installation 15	Discovery wizard
Description pane 172	creating 198
Details pane 169, 172	DiscoveryMain function 426
Details tab 108	DiscoveryProbe.properties file 38
Discovered CIs dialog box 119	domain credentials
Discovered CITs pane 236	references 177
discovery	Domain Scope Document
credential-less 308	blocking credentials 36

dictionary file 34	1
Domains and Probes pane 173	IBM DB2 Server
	discovery 297
E	IIS
Eclipse	discovery 370
configure to run Jython scripts in	importing data
debug mode 438	from external sources 373
Edit IP Range dialog box 164	troubleshooting 393
Edit Policy dialog box 166	Infrastructure wizard 128
Edit Probe Limitations for TQL Output dialog	Define Credentials page 130
box 126	Define IP Ranges page 128
Edit Related Probes dialog box 174	Preferences page 131
Edit Time Template dialog box 126	Schedule Discovery page 135
Edit Timetable dialog box 174	Summary page 135
errors	Input TQL Editor window 220 Input TQLs 50, 51
managing 78	installation
Execution Options pane 230	collector requirements 28
external resources 48	DDM Probe 28
external sources	on one machine 16
importing data from 373	procedure 15
importing data, troubleshooting 393	Integration - DDMi module 303
	Integration - NNM Layer 2 module 303
F	Integration - StorageEssential module 303
Find Discovery Resource dialog box 218	
Find Jobs dialog box 127	J
Find Text dialog box 220	
Framework instance 430	J2EE Port Scanning page
	J2EE wizard 138 J2EE wizard 136, 145
G	Define Credentials page 136
General Options pane 232	J2EE Port Scanning page 138
Global Configuration Files pane 237	JBoss page 143
globalFiltering.xml 206	Oracle Application Server page 144
globali litering.xiiii 200	Schedule Discovery page 144
	WebLogic page 139
Н	WebSphere page 141
hardening	Java exceptions
domain scope document (DSD)	handling 436
management 470	JBoss
hardware	protocol 180
requirements 28	JBoss page
host resources	J2EE wizard 143
discovery 315	JMX operation
HP Discovery and Dependency Mapping API	viewJobTriggeredCIs 87
Reference 410	viewJobTriggeredCIsWithErrorId 89

JMX operations	
activateJob 80	Manage Discovery Resources window 224
activateJobOnDestination 80	Microsoft Cluster Server
start/stop 81	discovery 292
viewJobErrorsSummary 81	Microsoft SQL Server
viewJobExecHistory 81	discovery 299
viewJobProblems 82	modeling.py 452
viewJobResultCiInstances 82	modules
viewJobResults 82	schedule to run 53
viewJobsStatuses 84	
viewJobStatus 85	NI.
job and pattern XML formats 453	N
Job Execution Policy pane 171	naming conventions 53
jobs	netutils.py 452
execution policies 159	network - basic 308
manually activating 52	network CI
running when job execution policy	manually creating 53
running 161	NNM protocol 180
viewing information through the JMX	NTCMD protocol 181
application 79	F
Jython	
configure Eclipse to run in debug	0
mode 438	oidToHostClass.xml 205
	Oracle
generating results 428	discovery 301
libraries and utilities 449 structure of the file 425	Oracle Application Server page
	J2EE wizard 144
using external Java JAR files 453	Oracle TNSName File Location page
	Database wizard 103
L	
Layer 2	D.
discovery 320	P
load balancers	packages 48
discovery 335	Pattern Management pane 228
logger.py 450	Pattern Signature pane 235
logs 54	patterns 47
changing log levels 55	assigning jobs to 420
Probe Gateway 59	creating 411
	defining input (Trigger CIT, Input
Probe Manager 60 severity levels 54	TQL) 412
	defining output 417
troubleshooting and limitations 61	development and testing 401
	finding correct credentials for
M	connections 434
Manage Discovery Resources 49, 195	implementing 411
Manage Discovery Resources 45, 193 Manage Discovery Resources user interface 208	modifying existing 405
wininge Discovery resources user interface 200	overriding parameters 418
	01

packaging and productization 402	collectors 28
scheduling 421	DDM Probe 28
separating 409	resource files 204
Trigger TQL 420	Result Grouping pane 234
writing new pattern 405	results
pattern-writing	filtering 35
introduction 396	Run Discovery 49
research stage 404	advanced mode workflow 74
Permission Editor dialog box 241	application 69
portNumberToPortName.xml 204	basic mode workflow 73
Preferences page	overview 70
Infrastructure wizard 131	user interface 91
Probe Gateway	view permissions 71
logs 59	
Probe Manager	S
logs 60	
Probe Selection pane 233	SAP
problem management 72	discovery 276
Properties tab 146	protocol 182
protocol	SAP JMX protocol 181
definitions 47	Schedule Discovery page
JBoss 180	Database wizard 104
NNM 180	Infrastructure wizard 135
NTCMD 181	J2EE wizard 144
SAP 182	Scope Definition dialog box 176
SAP JMX 181	script editor window 242
Siebel Gateway 183	script pane 243
SNMP 183	scripts 48
SQL 185	modifying out of the box 423
SSH 186	servers
Telnet 188	Business Availability Center 45
UDDI registry 190	Set Up Discovery Probes user interface 163
VMware Infrastructure 190	Set Up Discovery Probes window 176
WebLogic 191	shellutils.py 452
WebSphere 193	Show Results for Triggered CI page 152
WMI 194	Show Status Snapshot 49, 251
Protocol Parameters dialog box 175	(Job name) dialog box 253
protocols	Show Status Snapshot user interface 252
credentials 177	Show Status Snapshot window 254
	Siebel
R	discovery 282
	Siebel Gateway protocol 183
Ranges pane 172	SNMP protocol 183
Related CIs window 152	software
Relevant CITs pane 229	requirements 28
requirements	software elements

Software Library dialog box 248 Software Signature Editor dialog box 245 Solaris zones discovery 342 Source CIs dialog box 153 SQL protocol 185 SSH protocol 186 SSL configuring Discovery Probe 475 hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Solaris zones connection 63 Probe has disconnected status 65 results do not appear in map view 64 SAP Discovery fails 66 V Universal Description Discovery and Integration (UDDI) registry protocol 190 Used Scripts pane 240 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary JMX operations 81
Solaris zones discovery 342 Source CIs dialog box 153 SQL protocol 185 SSH protocol 186 SSL configuring Discovery Probe 475 hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 results do not appear in map view 64 SAP Discovery fails 66 SU Universal Description Discovery and Integration (UDDI) registry protocol 190 Used Scripts pane 240 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary MY operations 81
discovery 342 Source CIs dialog box 153 SQL protocol 185 SSH protocol 186 SSL configuring Discovery Probe 475 hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 SAP Discovery fails 66 U U U U Universal Description Discovery and Integration (UDDI) registry protocol 190 Used Scripts pane 240 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary MY operations 81
Source CIs dialog box 153 SQL protocol 185 SSH protocol 186 SSL
SQL protocol 185 SSH protocol 186 SSL
SSH protocol 186 SSL configuring Discovery Probe 475 hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Universal Description Discovery and Integration (UDDI) registry protocol 190 Used Scripts pane 240 V Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary MY operations 81
SSH protocol 186 SSL configuring Discovery Probe 475 hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Universal Description Discovery and Integration (UDDI) registry protocol 190 Used Scripts pane 240 V Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary MY operations 81
SSL configuring Discovery Probe 475 Integration (UDDI) registry protocol 190 Used Scripts pane 240 Start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Universal Description Discovery and Integration (UDDI) registry protocol 190 Used Scripts pane 240 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary MY operations 81
hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 registry protocol 190 Used Scripts pane 240 V Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary
hardening the Discovery Probe 475 start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 registry protocol 190 Used Scripts pane 240 V Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary MY operations 81
start/stop JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Used Scripts pane 240 V Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary
JMX operations 81 Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary
Statistics Results pane 117, 256 Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary IMV operations 81
Summary 145 J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary IMV operations 81
J2EE wizard 145 Summary page Database wizard 105 Infrastructure wizard 135 Veritas Cluster Server discovery 294 view permissions 71 viewJobErrorsSummary IMV operations 81
Summary page discovery 294 Database wizard 105 view permissions 71 Infrastructure wizard 135 MY operations 81
Database wizard 105 view permissions 71 Infrastructure wizard 135 viewJobErrorsSummary IMV operations 81
Infrastructure wizard 135 viewJobErrorsSummary
IMV operations 91
system components JMX operations of
handling deleted 196 viewJobExecHistory
JMX operations 81
viewJobProblems
JMX operations 82
Telnet protocol 188 viewJobResultCiInstances
Time Templates dialog box 153 JMX operations 82
TQL viewJobResults
building a view 76 JMX operations 82
defining 76 viewJobsStatuses
Trigger CIs 50, 51 JMX operations 84
Trigger CITs 50 viewJobStatus
Trigger TQL Editor window 154 JMX operations 85
00
00 ,
00
D (V 00
TTD (
Discovery tub imissing from tubs of
1.400
-
failure to connect to TTY agent 65
host name cannot be resolved to IP
address 63 not all networks and IPs discovered 64 WebLogic
discovery 204
not all TCF ports discovered 05
protocol 101
activation 62 protocol 191 webservice

```
discovery and dependency mapping
455
WebSphere
discovery 305
page in J2EE wizard 141
protocol 193
wizard
Database 99
J2EE 136
WMI protocol 194
wrapperProbe.log 63
```

Index