

HP Universal CMDB

for the Windows and Linux operating systems

Software Version: CP6.0, 9.00 or later

Discovery and Integrations Content

Document Release Date: June 2010

Software Release Date: June 2010



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 2005 - 2010 Hewlett-Packard Development Company, L.P

Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

AMD and the AMD Arrow symbol are trademarks of Advanced Micro Devices, Inc.

Google™ and Google Maps™ are trademarks of Google Inc.

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

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

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Acknowledgements

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

- This product includes OpenLDAP code from OpenLDAP Foundation (<http://www.openldap.org/foundation/>).
- This product includes GNU code from Free Software Foundation, Inc. (<http://www.fsf.org/>).
- This product includes JiBX code from Dennis M. Sosnoski.
- This product includes the XPP3 XMLPull parser included in the distribution and used throughout JiBX, from Extreme! Lab, Indiana University.
- This product includes the Office Look and Feels License from Robert Futrell (<http://sourceforge.net/projects/officeInfs>).
- This product includes JEP - Java Expression Parser code from Netaphor Software, Inc. (<http://www.netaphor.com/home.asp>).

Documentation Updates

The title page of this document 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

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 online support 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 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 register for an HP Passport ID, go to:

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

To find more information about access levels, go to:

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

Table of Contents

Welcome to This Guide	13
How This Guide Is Organized	13
Who Should Read This Guide	14
HP Universal CMDB Online Documentation	14
Additional Online Resources	15
Documentation Updates	16

PART I: INTRODUCTION 17

Chapter 1: Supported Content	19
Discovered Applications	20
Discovered Operating Systems	27
Chapter 2: General Information for Data Flow Content	29
Delete Files Copied to Remote Machine	29
Files Copied to a Remote Machine	30
Troubleshooting and Limitations	35

PART II: DISCOVERY CONTENT 37

Chapter 3: Load Balancers	39
Overview	39
Discover Load Balancers	40
Chapter 4: Microsoft Cluster	49
Discover Microsoft Cluster Servers	49

Chapter 5: Microsoft Network Load Balancing (NLB)	51
Microsoft NLB Overview	52
Discovery Mechanism	52
Discovering NLB with the Command Line Utility	53
Discover Microsoft Network Load Balancing Systems.....	55
MS NLB Cluster CIT	59
NLB Cluster Software CIT.....	61
Configuration File (NLB Port Rule)	62
Glossary	63
Components of the Network Load Balancing Architecture	64
Chapter 6: Veritas	67
Discover Veritas Cluster Servers	67
Chapter 7: DB2	73
Discover IBM DB2 Databases	73
Chapter 8: MS-SQL	77
Discovery by OS Credentials	77
Discover Microsoft SQL Server Database Application.....	78
Discover SQL Server by OS Credentials.....	80
Chapter 9: MySQL Replication Between Databases	83
Overview.....	83
Discover MySQL Configuration and Replication Jobs.....	84
Chapter 10: Oracle	93
Discover Oracle Databases.....	93
Discover Oracle Real Application Cluster (RAC).....	95
Chapter 11: Active Directory	107
Overview.....	107
Discover Active Directory Domain Controllers and Topology.....	108
Chapter 12: Microsoft Exchange	115
Overview.....	115
Discover Microsoft Exchange Server 2003	116
Discover Microsoft Exchange Server 2007	122
Discover Microsoft Exchange Server Topology with Active Directory.....	125
Chapter 13: Microsoft MQ (Message Queue)	135
Discover Microsoft MQ	135
Topology Discovery Methodology	139
Added Entities	149

Chapter 14: SAP	151
SAP Discovery Overview.....	152
Discover SAP ABAP	152
Discover SAP Solution Manager	157
Discover SAP Java	160
Troubleshooting and Limitations	162
Chapter 15: Siebel	163
Overview.....	163
Discover Siebel Topology	164
Troubleshooting and Limitations	170
Chapter 16: UDDI Registry	171
Overview.....	171
Discover UDDI Processes.....	172
Chapter 17: WebSphere MQ	175
Overview.....	176
Discover WebSphere MQ	178
Discovered CITs.....	182
Relationships	185
Enrichment Rule.....	188
Views and Reports	188
Troubleshooting and Limitations	192
Chapter 18: JBoss	195
JBoss Discovery Overview	195
Discover JBoss by JMX.....	196
Discover JBoss by Shell.....	198
Chapter 19: WebLogic	203
Discover J2EE WebLogic by JMX	203
Discover J2EE WebLogic by Shell.....	206
Troubleshooting and Limitations	210
Chapter 20: WebSphere	211
WebSphere Discovery Overview	211
Discover WebSphere by JMX	212
Discover WebSphere by Shell.....	214
Troubleshooting and Limitations	217
Chapter 21: Active and Passive Discovery Network Connections ..	219
Overview.....	219
Discover Processes	220
Discover TCP Traffic	227

Chapter 22: Network Advanced	229
Overview.....	229
Discover DNS Zones	230
Chapter 23: Network Basic	233
Network – Basic Overview.....	234
Network Workflow Overview.....	234
Discover Host Connection by Shell	235
Host Connection by SNMP	237
Discover Host Connection by WMI.....	240
Host Connection by Shell: Discover Windows Running F-Secure ...	244
Windows Processes.....	245
UNIX-Based Processes	247
Chapter 24: Credential-less	255
Overview.....	255
Discover Host Fingerprint with Nmap.....	256
Chapter 25: Host Resources and Applications	263
Host Resources and Applications Overview.....	263
Host Resources and Applications – Workflow	266
Revert to Previous Method of Discovering Installed Software	274
Troubleshooting and Limitations	275
Chapter 26: Layer 2	277
Overview.....	277
Discover Layer 2 Objects	278
Chapter 27: Discovery Tools	295
Overview.....	295
Troubleshooting and Limitations	296
Chapter 28: Importing Data from External Sources	297
Importing Data from External Sources – Overview	298
The External_source_import Package.....	300
The Import from CSV File Job.....	301
The Import from Properties File Job	305
The Import from Database Job.....	306
The External Source Mapping Files.....	311
Converters	312
Import CSV Data from an External Source – Scenario.....	314
Troubleshooting and Limitations	318
Chapter 29: VMware	321
Discover VMware Infrastructure Topology	321
Discover VMware VMotion	343

Chapter 30: Apache Tomcat	347
Overview.....	348
Discover Apache Tomcat.....	349
Discover Bugzilla, Wordpress, and MediaWiki	353
Chapter 31: Microsoft Internet Information Services (IIS)	355
Discover Microsoft Internet Information Services (IIS) – Previous Topology	355
Discover Microsoft Internet Information Services (IIS) – Current Topology	358

PART III: SUPPORTED INTEGRATIONS 367

Chapter 32: Network Node Manager <i>i</i> (NNMi) Integration with HP Universal CMDB	369
NNMi Integration Overview	370
NNMi - UCMDB Integration Architecture	371
Set Up HP NNMi–HP UCMDB Integration	372
Run HP NNMi–UCMDB Integration	373
Use the HP NNMi–HP UCMDB Integration	381
Change the HP NNMi–HP UCMDB Integration Configuration	384
Disable HP NNMi–HP UCMDB Integration Configuration	384
Perform Impact Analysis	385
HP NNMi–HP UCMDB Integration Configuration Form Reference	385
NNMi Protocol Connection Parameters	389
Troubleshooting and Limitations	390
Chapter 33: Storage Essentials (SE) Integration with HP Universal CMDB	393
SE Integration – Overview	393
Discover the SE Oracle Database	395
Storage Essentials Integration Packages	398
Discovered CITs.....	398
Views.....	403
Correlation Rules.....	407
Reports.....	409

Chapter 34: EMC Control Center (ECC) Integration with HP Universal CMDB413	
ECC Integration – Overview	414
Discover the ECC Storage Topology	415
ECC Job SQL Queries.....	422
Views	425
Correlation Rules.....	429
Reports.....	431
Chapter 35: Data Dependency and Mapping Inventory Integration with HP Universal CMDB435	
Overview.....	436
DDMi Adapter	436
Populate the CMDB with Data from DDMi.....	438
Federate Data with DDMi.....	441
Customize the Integration Data Model in UCMDB	441
DDMi Adapter Configuration Files	443
Troubleshooting and Limitations	444
Chapter 36: Microsoft SCCM/SMS Integration with HP Universal CMDB 445	
SCCM/SMS Integration – Overview	446
SMS Adapter	447
Populate the CMDB with Data from SCCM/SMS	449
Federate Data with SCCM/SMS	453
Customize the Integration Data Model in Universal CMDB.....	454
SCCM/SMS Integration Package.....	455
SMS Adapter Configuration Files	458
Troubleshooting and Limitations	459
Index 461	

Welcome to This Guide

This guide explains how to bring data into HP Universal CMDB either through discovery or integration.

This chapter includes:

- ▶ How This Guide Is Organized on page 13
- ▶ Who Should Read This Guide on page 14
- ▶ HP Universal CMDB Online Documentation on page 14
- ▶ Additional Online Resources on page 15
- ▶ Documentation Updates on page 16

How This Guide Is Organized

The guide contains the following chapters:

Part I Introduction

Includes supported discovery components and general information for Discovery and Integrations content.

Part II Discovery Content

Includes detailed information on discovering system entities.

Part III Supported Integrations

Describes how to retrieve data from other products and include the data in the UCMDB.

Who Should Read This Guide

This guide is intended for the following users:

- HP Universal CMDB administrators
- HP Universal CMDB platform administrators
- HP Universal CMDB application administrators
- HP Universal CMDB data collector administrators

Readers of this guide should be knowledgeable about enterprise system administration, have familiarity with ITIL concepts, and be knowledgeable about HP Universal CMDB.

HP Universal CMDB Online Documentation

HP Universal CMDB includes the following online documentation:

Readme. Provides a list of version limitations and last-minute updates. From the HP Universal CMDB DVD root directory, double-click **readme.html**. You can also access the most updated readme file from the HP Software Support Web site.

What's New. Provides a list of new features and version highlights. In HP Universal CMDB, select **Help > What's New**.

Printer-Friendly Documentation. Choose **Help > UCMDB Help**. The following guides are published in PDF format only:

- **Deployment.** Explains the hardware and software requirements needed to set up HP Universal CMDB, how to install or upgrade HP Universal CMDB, how to harden the system, and how to log in to the application.
- **Database.** Explains how to set up the database (MS SQL Server or Oracle) needed by HP Universal CMDB.
- **Discovery and Integrations Content.** Explains how to run discovery to discover applications, operating systems, and network components running on your system. Also explains how to discover data on other data stores through integration.

HP Universal CMDB Online Help includes:

- ▶ **Modeling.** Enables you to manage the content of your IT Universe model.
- ▶ **Data Flow Management.** Explains how to integrate HP Universal CMDB with other data stores and how to set up HP Universal CMDB to discover network components.
- ▶ **Administration.** Explains how to work with HP Universal CMDB.
- ▶ **Developer Reference.** For users with an advanced knowledge of HP Universal CMDB. Explains how to define and use adapters and how to use APIs to access data.

Online Help is also available from specific HP Universal CMDB windows by clicking in the window and clicking the **Help** button.

Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site (www.adobe.com).

Additional Online Resources

Troubleshooting & Knowledge Base. Enables you to search the Self-solve knowledge base in the Troubleshooting page on the HP Software Support Web site. Choose **Help > Troubleshooting & Knowledge Base**. The URL for this Web site is <http://h20230.www2.hp.com/troubleshooting.jsp>.

HP Software Support. Enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > HP Software Support**. The URL for this Web site is www.hp.com/go/hpssoftwaresupport.

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 user ID, go to:

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

HP Software Web site. Provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help > HP Software Web site**. The URL for this Web site is www.hp.com/go/software.

Documentation Updates

HP Software is continually updating its product documentation with new information.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to the HP Software Product Manuals Web site (<http://h20230.www2.hp.com/selfsolve/manuals>).

Part I

Introduction

1

Supported Content

This chapter includes:

Reference

- ▶ Discovered Applications on page 20
- ▶ Discovered Operating Systems on page 27

Reference

Discovered Applications

Note: Additional supported content is also publicly available to download through the HP Live Network (<https://h20090.www2.hp.com/>). Follow the **Content Pack Documentation and Content Packs** link. You will need an HP Passport user name and password.

Vendor	Product	Versions	Credentials	Discovers...
Apache	Http Server	1.3, 2.0, 2.2	Shell	Apache Http server Listening ports, Virtual hosts, configuration files, Web application, Apache Modules (including mod_proxy and mod_proxy_balancer)
Apache	Tomcat	5.5, 6.x	Shell	Tomcat Server, Web applications, configuration files, virtual servers, listening ports, Tomcat Cluster, Tomcat Service
BEA	Weblogic Application Server	6.x, 7.x, 8.x, 9.x, 10.x	Shell	Weblogic J2EE Server, J2EE application, JDBC datasource, Database, EJB Module, Web Module and JMS resources, J2EE Domain, J2EE Cluster

Vendor	Product	Versions	Credentials	Discovers...
Cisco	CSS	6.10, 7.4	SNMP	<p>Mapping of Virtual IPs to real IP addresses of servers configured for load balancing; configuration files, load balancing algorithms, and end user IP addresses</p> <p>Note: Cisco WebNS is the software version running on the 11000 and 11500 series CSS</p>
EMC	EMC Control Center (ECC)	6.x	Oracle DB	<p>Synchronized Configuration Items (CIs) currently include Storage Arrays, Fiber Channel Switches, Hosts (Servers), Storage Fabrics, Storage Zones, Logical Volumes, Host Bus Adapters, Storage Controllers, and Fiber Channel Ports. Integration also synchronizes physical relationships between various hardware and logical relationships between Logical Volumes, Storage Zones, Storage Fabrics, and hardware devices to enable end-to-end mapping of the storage infrastructure in UCMDB.</p> <p>Note: Synchronized Content, not discovery of application topology</p>
F5	BIG-IP LTM	4.6, 9.1	SNMP	<p>Mapping of Virtual IPs to real IP addresses of servers configured for load balancing; configuration files, load balancing algorithms, and end user IP addresses</p>

Vendor	Product	Versions	Credentials	Discovers...
HP	Network Node Manager (NNM)	8.1	NNM API	Discovered nodes, IPs, networks, interfaces and Layer 2 connection information to create a Layer 2 topology in UCMDB
HP	ServiceGuard		Shell	SG cluster software, SG packages, SG resources, cluster members
HP	Storage Essentials (SE)	6.x	SQL	Synchronized Configuration Items (CIs) including Storage Arrays, Fiber Channel Switches, Hosts (Servers), Storage Fabrics, Storage Zones, Logical Volumes, Host Bus Adapters, Storage Controllers, and Fiber Channel Ports. The integration also synchronizes physical relationships between various hardware and logical relationships between Logical Volumes, Storage Zones, Storage Fabrics, and hardware devices to enable end-to-end mapping of the storage infrastructure in UCMDB
IBM	DB2 Universal Database (UDB)	8.2, 9.1, 9.5	SQL	<p>DB2 databases, including instances, tablespaces, users, processes, jobs (backup routines, log routines, and so on), any database objects</p> <p>Discovery through:</p> <ul style="list-style-type: none"> ▶ direct connection to DB2 database, ▶ SQL queries ▶ HP DFM z/OS Mainframe <p>Note: Discovery Agent, 9.2, 9.5 are recent versions</p>

Vendor	Product	Versions	Credentials	Discovers...
IBM	HTTP Server	5, 6.1	Shell	IBM Http Server's WebSphere plug-in configuration by parsing the IHS plug-in configuration file
IBM	MQ Series (aka WebSphere MQ)	5.31, 6		<p>MQ subsystems at the system configuration level; DFM does not monitor or discover which active jobs or applications are running through the queues.</p> <p>Discovery includes Queue Managers, System Parameters, Queue-Sharing Groups, related DB2 Data-Sharing Groups, Cross Coupling Facility groups/members, Channel Initiator, Sender Channel, Server Channel, Receiver Channel, Requester Channel, Client Connection Channel, Server Connection Channel, Cluster Sender Channel, Cluster Receiver Channel, Alias Queue, Model Queue, Local Queue, Transmission Queue, Remote Queue, MQ Process, and MQ Cluster.</p>
IBM	WebSphere Application Server	5.x, 6.x, 7.x	Shell	J2EE Server, J2EE application, JDBC datasource, Database, EJB Module, Web Module, J2EE Domain and JMS resources
JBoss	Application Server	3.x, 4.x, 5.x	Shell or JMX	JBoss J2EE application server, EJB Module, Entity Bean, J2EE Application, J2EE Domain, JDBC Data Source, JMS Destination, JMS Server, JVM, Message Driven Bean, Servlet, Session Bean, Web module

Vendor	Product	Versions	Credentials	Discovers...
Microsoft	Active Directory	2003	LDAP	Domain Controllers, Forest, Domain, Site, and Trusts
Microsoft	Active Directory Server	2000, 2003, 2008	LDAP	Forest, Sites, Sitelinks, Domain controllers, Networks, and so on
Microsoft	Cluster Services	Windows Server 2000, Windows Server 2003	Shell	Cluster software, configuration files, cluster members, MCS Resource Groups, MCS Resources
Microsoft	Exchange Server	2003	WMI	Administrative Group, Directory Service Access DC, Exchange Folder, Exchange Folder Tree, Exchange Links, Exchange Message Queue, Exchange System, Routing Group
Microsoft	Exchange Server	2003, 2007	LDAP	Forest, Sites, Exchange folders, folder trees, Administrative groups, Connectors
Microsoft	Exchange Server	2007	NTCMD (PowerShell)	Exchange Server, Exchange roles
Microsoft	IIS	6,7	Shell	Discover the IIS Web Server, IIS Web Site, IIS virtual Dir, IIS Application pool, web services and configuration files
Microsoft	SQL Server	7, 2000, 2005	SQL	Discovery of MS SQL databases, including instances, tablespaces, users, processes, jobs (backup routines, log routines, and so on), any database objects, MS SQL clustering, and log file shipping tasks 2008 support?

Vendor	Product	Versions	Credentials	Discovers...
MySQL	MySQL Database	3.x, 4.x, 5.x	Shell	Support MySQL Master-Master and Master-Slave configuration. Discover MySQL Database, configuration files, Replication job
Nortel	Alteon		SNMP	Mapping of Virtual IPs to real IP addresses of servers configured for load balancing; configuration files, load balancing algorithms, and end user IP addresses
Oracle	Database (including RAC)	9,10g,11g	Shell	Oracle database, TNS Listener software, and Oracle RAC
Oracle	Database (plus RAC)	8, 9, 10g	SQL	Oracle databases, including SIDs, TNS names, instances, tablespaces, users, processes, jobs (backup routines, ONP, jobs, log routines, and so on), and any database objects
Oracle	E-Business Suite	11i, 12	SQL	Oracle E-Business applications, such as Oracle Financials; infrastructure components, Web servers, application servers, individual components, and configuration files
Oracle	Siebel CRM	7.5, 7.7	Shell	Discovery of Siebel Enterprise, including Siebel applications (CallCenter, Financial, and so on), Siebel infrastructure components, Siebel Web servers, application servers, gateway servers, individual Siebel, components and configuration files

Chapter 1 • Supported Content

Vendor	Product	Versions	Credentials	Discovers...
Oracle	WebLogic	8.x, 9.x, 10.x	Shell or JMX	
SAP	NetWeaver	2.x, 4, 7	JMX; SAP JCo	SAP ABAP Application Server, SAP Clients, SAP Gateway, SAP System, SAP Work Process, JDBC Data Sources, Databases, Hosts in deployment with IPs, SAP J2EE Application Server, SAP J2EE Dispatcher, SAP J2EE Server Process, SAP J2EE Central Services, J2EE domain, EJBs, EJB Modules, Entity Beans, Stateful/Stateless Session Beans, Web Module, SAP Business Process, SAP Business Scenario, SAP Process Step, SAP Project, SAP Transaction, SAP Application Components, SAP Transports, SAP ITS AGate, SAP ITS WGate
SAP	SAP Solution Manager	6.4, 7.0	SAP JCo	SAP ABAP Application Server, SAP Clients, SAP System, JDBC Data Sources, Databases, SAP J2EE Application Server, SAP J2EE Dispatcher, SAP J2EE Central Services, J2EE domain
Sun	MySQL Database Server	4.x and above	Shell	MySQL databases and MySQL replication topology
Sun	Solaris Zones	5.1	Shell	Containers, zones, and share resources
Sybase	Adaptive Server Enterprise	10.x, 11.x, 12.x, 15.x	SQL	Sybase databases, including instances, tablespaces, users, processes, jobs (backup routines, log routines, and so on), and any database objects

Vendor	Product	Versions	Credentials	Discovers...
Symantec	Veritas Cluster Server (VCS) for UNIX	2.x, 3.x, 4.x, 5.x	Shell	Cluster Software, configuration files, cluster members, VCS Resource Groups, VCS Resources
Tomcat	Apache	5.x, 6.x	Shell	Tomcat Server instances, Web applications, configuration files, virtual servers, listening ports
VMware	ESX	2.5, 3	Shell	
VMware	ESX & ESXi	2.5, 3, 3i, 3.5	VIM	ESX servers, cluster groups, virtual resource groups
VMware	vCenter (formerly Virtual Center)	2.01, 2.5	VIM and WMI	Virtual Center Server, License Server, ESX servers, cluster groups, virtual resource groups

Discovered Operating Systems

Vendor	Product	Versions	Credentials	Content
IBM	AIX	5.x, 6.x		OS, Memory, Disks, CPU, Processes, Software (packages), Services (daemons), Files, Directories, Local Users
HP	HP-UX	10.xx, 11.xx		OS, Memory, Disks, CPU, Processes, Software (packages), Services (Daemons), Files, Directories, Local Users, HP-UX Clusters
IBM	OS/390		SNMP	Simple mainframe discovery identifies Sysplex, LPARs, and IPs
RedHat	RedHat Enterprise Linux			OS, Memory, Disks, CPU, Processes, Software (packages), Services (daemons), Files, Directories, Local Users

Chapter 1 • Supported Content

Vendor	Product	Versions	Credentials	Content
Sun	Solaris	5.9, 5.10		OS, Memory, Disks, CPU, Processes, Software (packages), Services (daemons), Files, Directories, Local Users
Microsoft	Windows	All Versions		OS, Memory, Disks, CPU, Processes, Software, Services, Files, Directories, Local Users

2

General Information for Data Flow Content

This chapter includes:

Tasks

- ▶ Delete Files Copied to Remote Machine on page 29

Reference

- ▶ Files Copied to a Remote Machine on page 30

Troubleshooting and Limitations on page 35

Tasks

Delete Files Copied to Remote Machine

During discovery, Data Flow Probe copies files to a remote Windows machine. For details, see "Files Copied to a Remote Machine" on page 30.

To configure DFM to delete files copied to the destination machine, once discovery is finished:

- 1** Access the `globalSettings.xml` file: **Adapter Management > AutoDiscoveryContent > Configuration Files.**
- 2** Locate the `removeCopiedFiles` parameter.
 - ▶ **true.** The files are deleted.
 - ▶ **false.** The files are not deleted.
- 3** Save the file.

To control xCmd behavior:

- 1** In the **globalSettings.xml** file, locate the **NtcmdAgentRetention** parameter.
- 2** Enter one of the following:
 - ▶ **0.** (The default) Unregister the service and delete the remote executable file. (**Unregister:** stop the service and remove it from the remote machine, so that it is no longer listed in the list of services.)
 - ▶ **1.** Unregister the service, but leave the executable file on the file system.
 - ▶ **2.** Leave the service running, and leave the executable file on the file system.

Reference

Files Copied to a Remote Machine

This section lists the files that the Data Flow Probe copies to a remote Windows machine, to enable discovery of the machine's components.

DFM copies the files to the `%SystemRoot%\system32\drivers\etc\` folder on the remote machine.

Note:

- ▶ DFM runs **xCmdSvc.exe** to connect to and retrieve the Shell on the remote machine.
 - ▶ When the **wmic** command is launched on the remote Windows machine, by the **Host Connection by Shell** or **Host Resources and Applications by Shell** jobs, an empty **TempWmicBatchFile.bat** file is created.
-

This section includes the following topics:

- "adsutil.vbs" on page 31
- "getfilever.vbs" on page 31
- "reg_mam.exe" on page 31
- "meminfo.exe" on page 32
- "diskinfo.exe" on page 32
- "processlist.exe" on page 32
- "Exchange_Server_2007_Discovery.ps1" on page 33
- "GetFileModificationDate.vbs" on page 33
- "junction.exe" on page 34

adsutil.vbs

The Visual Basic script used for discovery of Microsoft IIS applications.

DFM copies this Visual Basic script to the remote machine to discover IIS.

Relevant DFM Job: IIS Applications by NTCMD

Content Pack Version: All

getfilever.vbs

The Visual Basic script is used to identify the version of the running software. The script retrieves the executable or DLL file version on Windows machines.

This Visual Basic script is used by Shell-based application signature plug-ins to retrieve the version of a particular software on the remote machine.

Relevant DFM Job: Host Resources and Applications by Shell

Content Pack Version: All

reg_mam.exe

The copy of the Microsoft **reg.exe** file that enables querying the registry.

If DFM does not discover a native **reg.exe** file, this executable is copied to the remote Windows machine. This situation occurs with some previous Windows versions (for example, Windows 2000) where the tool is not included by default but can still function there correctly.

Relevant DFM Job: Host Resources and Applications by Shell

Content Pack Version: All

meminfo.exe

The executable that enables the retrieval of memory information.

DFM discovers memory information with the **wmic** query. However, if the **wmic** query fails to execute, DFM copies the **meminfo.exe** file to the remote machine. This failure can occur if, for example, **wmic.exe** is not included in the **PATH** system variable or is completely absent on the remote machine, as is the case on Windows 2000.

Relevant DFM Job: Host Resources and Applications by Shell

Content Pack Version: All

diskinfo.exe

The executable that enables the retrieval of disk information when it is not available to be retrieved by **wmic**.

DFM discovers default disk information with the **wmic** query. However, if the **wmic** query fails to execute, DFM copies the **diskinfo.exe** file to the remote machine. This failure can occur if, for example, **wmic.exe** is not included in the **PATH** system variable or is completely absent on the remote machine, as is the case on Windows 2000.

Relevant DFM Job: Host Resources and Applications by Shell

Content Pack Version: All

processlist.exe

The executable that enables the retrieval of process information together with command line, PID and other relevant information.

DFM discovers default process information with the **wmic** query. However, if the **wmic** query fails to execute, DFM copies the **processlist.exe** file to the remote machine. This failure can occur if, for example **wmic.exe** is not included in the **PATH** system variable or is completely absent on the remote machine, as is the case on Windows 2000.

Relevant DFM Job: Host Resources and Applications by Shell

Content Pack Version: All

Exchange_Server_2007_Discovery.ps1

The PowerShell script for MS Exchange 2007 discovery.

DFM uses a PowerShell scenario to discover Microsoft Exchange 2007 by NTCMD. This file, therefore, must be copied to the remote machine.

Relevant DFM Jobs:

- Microsoft Exchange Connection by NTCMD
- Microsoft Exchange Topology by NTCMD

Content Pack Version: CP4

GetFileModificationDate.vbs

The Visual Basic script for retrieving the file modification date (disregarding locale).

The most common use case is when DFM must retrieve the last modification date of a configuration file of a discovered application.

Relevant DFM Jobs:

- Apache Tomcat by Shell
- File Monitor by Shell
- IIS Applications by NTCMD
- IHS Websphere Plugin by Shell
- J2EE Weblogic by Shell
- J2EE WebSphere by Shell or JMX

- ▶ J2EE WebSphere by Shell
- ▶ Oracle TNSName by Shell
- ▶ SAP Profiles by Shell
- ▶ SAP System By Shell
- ▶ Service Guard Cluster Topology by TTY
- ▶ Siebel Application Server Configuration
- ▶ Software Element CF by Shell
- ▶ Veritas Cluster by Shell
- ▶ Webserver by Shell

Content Pack Version: CP5

junction.exe

This executable file, part of the Sysinternals Suite (<http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>), enables the creation of a junction point. DFM uses this file if the **linkd.exe** and **mklink.exe** tools are absent on the remote machine.

When DFM runs discovery on a Windows x64 machine, DFM needs to bypass the Windows redirect feature running on that machine. DFM does this by creating a link to the **%SystemRoot%\System32** folder with either the **linkd.exe** or **mklink.exe** tool. However, if these tools are missing on the remote machine, DFM transfers **junction.exe** to the remote machine. DFM is then able to launch the 64-bit version of the system executable files. (Without this 64-bit version, DFM would be locked into an isolated 32-bit world.)

Note: This junction point is automatically removed once discovery is complete.

Relevant DFM Jobs:

- Host Resources and Applications by Shell
- Microsoft Exchange Connection by NTCMD
- Microsoft Exchange Topology by NTCMD

Content Pack Version: CP5

Troubleshooting and Limitations

- Following the run of the **Host Connection by SNMP** or **Host Networking by SNMP** jobs, many warning messages are displayed:

```
Detected multiple updates in bulk - found attribute: 'interface_description' on current
CIT: 'interface'
```

These messages can be safely ignored. To prevent the messages being displayed, you can change the **multipleUpdateIgnoreTypes** parameter in the **GlobalSettings.xml** file:

```
<!--
    multipleUpdateIgnoreTypes - don't check multiple updates for the following
    types
-->
<property
name="multipleUpdateIgnoreTypes">process,clientserver,host</property>
```

Add the **interface** CIT to this list of CITs to be ignored.

- When running the **Host Connection by NTCMD** job, the following error may be displayed:

```
Error: Multiple connections to a server or shared resource by the same user, using
more than one user name, are not allowed.
```

This may be caused by one of the following NetBIOS protocol limitations:

- ▶ The network share is considered to be in use even though it is not, that is, the session is frozen. In this case, try the following command:

```
net use * /delete
```

- ▶ The network share is in use by another user whose user name is bound to the local machine user name. In this case, you can reconfigure the remote machine security policy, or wait for the other user to finish working.

Part II

Discovery Content

3

Load Balancers

This chapter includes:

Concepts

- ▶ Overview on page 39

Tasks

- ▶ Discover Load Balancers on page 40

Concepts

Overview

DFM 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)

Tasks

Discover Load Balancers

This task explains how to discover load balancers and includes the following steps:

- "Supported Versions" on page 40
- "Prerequisites" on page 41
- "Discovery Workflow" on page 42
- "Load Balancer CITs" on page 42
- "The Load_balancing Package" on page 43
- "The Alteon_application_switch Package" on page 44
- "The F5_BIGIP_LTM Package" on page 45
- "The Cisco_CSS Package" on page 46
- "Topology Map" on page 47

1 Supported Versions

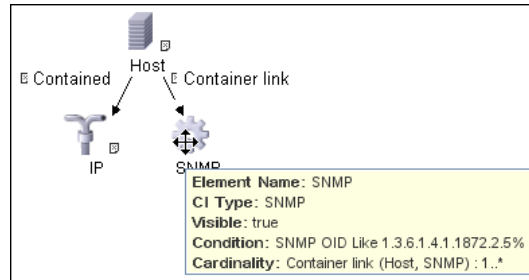
The supported version for each load balancer is as follows:

- F5 BIG-IP Local Traffic Manager, versions 9 and 4.
- Nortel Application Switches. No known limitations
- Cisco Content Services Switches. No known limitations.

2 Prerequisites

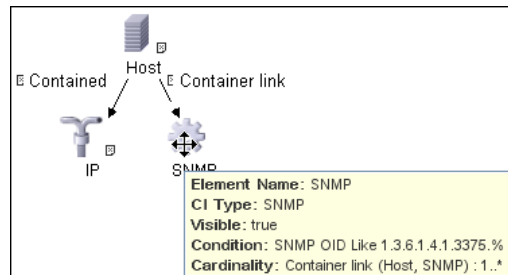
Run the **Host Connection by SNMP** job to discover and create SNMP CIs which answer the following requirements:

- To be the trigger TQL for the **Alteon application switch by SNMP** job with the following condition:



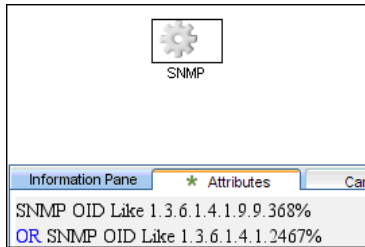
SNMP OID Like 1.3.6.1.4.1.1872.2.5%

- 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%

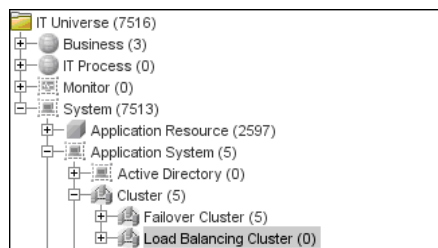
3 Discovery Workflow

- ▶ **Host Connection by SNMP.** For details on the prerequisites to running a load balancer job, see "Prerequisites" on page 41.
- ▶ Any of the following jobs:
 - ▶ **F5 BIG-IP LTM by SNMP**
 - ▶ **Alteon application switch by SNMP**
 - ▶ **Cisco CSS by SNMP**

4 Load Balancer CITs

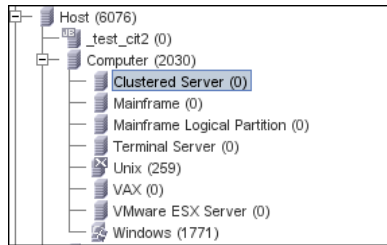
The following CITs model load balancer topology:

Load Balancer Software:



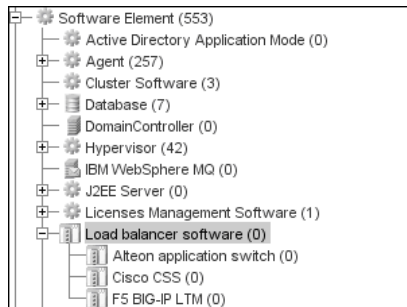
This CIT represents software that provides load balancing solutions. For details on the supported load balancers, see "Overview" on page 39.

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

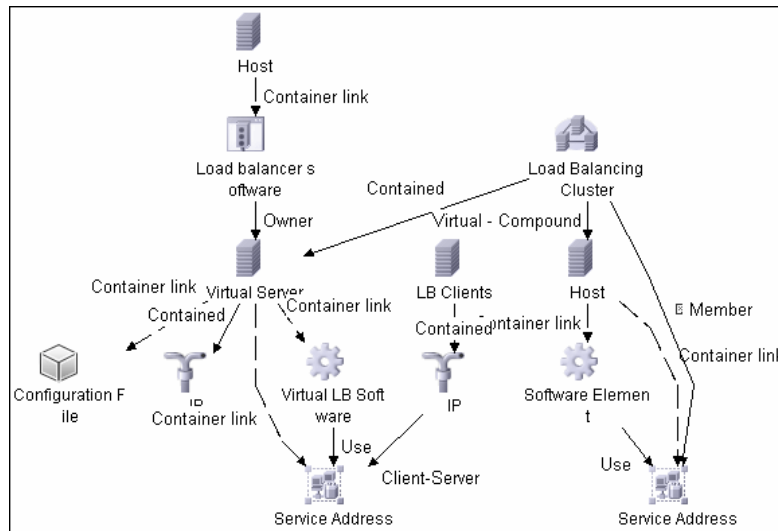
This package contains data that is common to all load balancer solutions.

The Network – Load balancer module contains the load balancing jobs:

- Alteon application switch by SNMP
- Cisco CSS by SNMP
- F5BIG-IP LTM by SNMP

Note: The Network – Load Balancer module is located under the Legacy module.

The following view displays the topology (**View Manager > Root > Application > Load balancer**):



6 The Alteon_application_switch Package

This package contains a class model definition, an adapter, and a job used to discover Nortel application switches by SNMP.

To run this package, activate the **Alteon application switch by SNMP** job. DFM discovers Nortel (Alteon) load balancers and all related CIs.

The following SNMP tables are queried:

Table Name	Name From MIB	OID
Virtual servers	slbCurCfgVirtServer Table	1.3.6.1.4.1.1872.2.5.4.1.1.4.2.1
Virtual services	slbCurCfgVirtServices Table	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	slbCurCfgRealServer Table	1.3.6.1.4.1.1872.2.5.4.1.1.2.2.1
Port links	slbCurCfgRealServPort Table	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, an adapter, 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. DFM 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	ItmVirtualServTable	1.3.6.1.4.1.3375.2.2.10.1.2.1
Pools	ItmPoolTable	1.3.6.1.4.1.3375.2.2.5.1.2.1
Pools to server	ItmVirtualServPool Table	1.3.6.1.4.1.3375.2.2.10.6.2.1
Pool members	ItmPoolMemberTable	1.3.6.1.4.1.3375.2.2.5.3.2.1

Table Name	Name From MIB	OID
Rules to servers	ltmVirtualServRule Table	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, an adapter, and a job used to discover Cisco Content Services Switches by SNMP. This package supports all versions of Cisco CSS.

To run this package, activate the **Cisco CSS by SNMP** job. DFM 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):

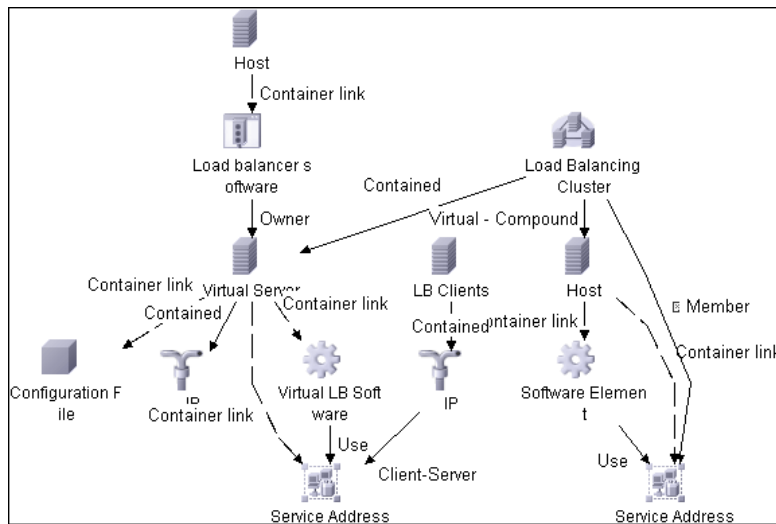
- **apCntTable** provides information about virtual addresses, virtual services, and pools.
- **apSvcTable** provides information about physical hosts included in the pool.

► **apCnsvTable** describes which host is included in which pool.

apSvcTable can contain entries for which there is no corresponding row in **apCnsvTable**. 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	apCnsvEntry	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



4

Microsoft Cluster

This chapter includes:

Tasks

- ▶ Discover Microsoft Cluster Servers on page 49

Tasks

Discover Microsoft Cluster Servers

The MS Cluster discovery process enables you to discover the topology of a Microsoft Cluster Server on the network.

This task includes the following steps:

- ▶ "Network and Protocols" on page 49
- ▶ "Discovery Workflow" on page 50
- ▶ "Topology Map" on page 50

1 Network and Protocols

- ▶ **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.

2 Discovery Workflow

In the Discovery Control Panel window, activate the modules in the following order:

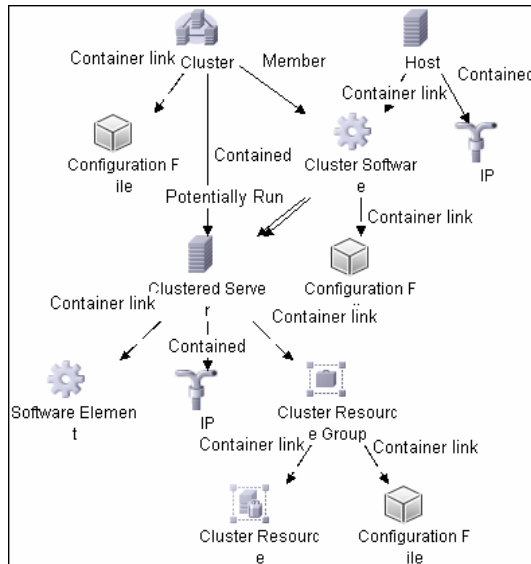
- **Network – Basic** (Host Connection by Shell)
- **Network – Host Resources and Applications**
- **Cluster – Microsoft Cluster** (MS Cluster by NTCMD)

For details on the CIs that are discovered, see the Statistics table in the Details tab.

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**, it is assumed that different, dedicated, virtual IPs are being assigned; these IPs are configured for the use of the cluster clients.



5

Microsoft Network Load Balancing (NLB)

Note: This functionality is available as part of Content Pack 6.00 or later.

This chapter includes:

Concepts

- ▶ Microsoft NLB Overview on page 52
- ▶ Discovery Mechanism on page 52
- ▶ Discovering NLB with the Command Line Utility on page 53

Tasks

- ▶ Discover Microsoft Network Load Balancing Systems on page 55

Reference

- ▶ MS NLB Cluster CIT on page 59
- ▶ NLB Cluster Software CIT on page 61
- ▶ Configuration File (NLB Port Rule) on page 62
- ▶ Glossary on page 63
- ▶ Components of the Network Load Balancing Architecture on page 64

Concepts

Microsoft NLB Overview

Network Load Balancing (NLB) distributes IP traffic to multiple copies (or instances) of a TCP/IP service, such as a Web server, each running on a host within the cluster. NLB transparently partitions the client requests among the hosts and lets the clients access the cluster using one or more virtual IP addresses. From the client's point of view, the cluster appears to be a single server that answers these client requests. Each server receives all client requests, but NLB decides which server should respond.

Discovery Mechanism

DFM triggers on Windows machines with more than one (two or more) IP addresses, and collects information using the **nlb.exe** command line utility. (In earlier versions of the Windows 2000 family, **wlbs.exe** is used.) These utilities enable the retrieval of all NLB-related information. For details, see "Discovering NLB with the Command Line Utility" on page 53.

There is no need for DFM to collect information from every participating node to verify that an MS NLB cluster system exists: even one single machine running the software is considered a cluster machine. If more machines are discovered that include the NLB service (with the same settings as the first machine), the NLB cluster begins the convergence process.

Furthermore, cluster information is collected by discovering one node at a time because nodes participating in a cluster do not include information about the other participants.

Discovering NLB with the Command Line Utility

The **nlb.exe** command line utility runs with the **params** key and outputs information about all NLB clusters on a discovered machine.

- If NLB is not installed on a Windows 2003 Server machine, the output is as follows:

```
WLBS Cluster Control Utility V2.4 (c) 1997-2003 Microsoft Corporation.  
WLBS is not installed on this system or you do not have sufficient privileges to  
administer the cluster.
```

- If an NLB cluster is set up on the machine, the output is as follows:

```
Cluster 192.168.0.222
Retrieving parameters
Current time      = 9/3/2009 1:02:38 PM
HostName         = ddmvm-2k3-s
ParametersVersion = 4
CurrentVersion   = 00000204
EffectiveVersion = 00000201
InstallDate      = 4A9E51F5
HostPriority      = 1
ClusterIPAddress = 192.168.0.222
ClusterNetworkMask = 255.255.255.0
DedicatedIPAddress = 192.168.0.2
DedicatedNetworkMask = 255.255.255.0
McastIPAddress   = 0.0.0.0
ClusterName      = cluster2.domain.com
ClusterNetworkAddress = 03-bf-c0-a8-00-de
IPToMACEnable    = ENABLED
MulticastSupportEnable = ENABLED
IGMPSupport      = DISABLED
MulticastARPEnable = ENABLED
MaskSourceMAC    = ENABLED
AliveMsgPeriod   = 1000
AliveMsgTolerance = 5
NumActions       = 100
NumPackets       = 200
NumAliveMsgs     = 66
DescriptorsPerAlloc = 512
MaxDescriptorAllocs = 512
TCPConnectionTimeout = 60
IPSecConnectionTimeout = 86400
```

```

FilterICMP          = DISABLED
ClusterModeOnStart = STARTED
HostState           = STARTED
PersistedStates     = NONE
ScaleSingleClient   = DISABLED
NBTSupportEnable    = ENABLED
NetmonAliveMsgs     = DISABLED
IPChangeDelay       = 60000
ConnectionCleanupDelay = 300000
RemoteControlEnabled = DISABLED
RemoteControlUDPPort = 2504
RemoteControlCode   = 00000000
RemoteMaintenanceEnabled = 00000000
BDATeaming          = NO
TeamID              =
Master               = NO
ReverseHash          = NO
IdentityHeartbeatPeriod = 10000
IdentityHeartbeatEnabled = ENABLED

```

PortRules (1):

VIP	Start	End	Prot	Mode	Pri	Load	Affinity
All	0	65535	Both	Multiple		Eq	Single

No special rules are used for mapping the output to the CITs; all CI attributes repeat the output data names. Data is verified by comparing it to cluster nodes that have already been discovered.

Tasks



Discover Microsoft Network Load Balancing Systems

This task includes the following steps:

- "Network and Protocols" on page 56
- "Discovery Workflow" on page 56
- "Packages" on page 56

- "Discovered CIs" on page 57
- "Trigger TQL" on page 58
- "MS NLB by NTCMD Adapter" on page 58
- "Views" on page 59
- "Topology" on page 59

1 Network and Protocols

- **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.

Verify that the user defined in the NTCMD protocol is granted administration rights for Shell execution on the remote machine.

The NTCmd protocol retrieves information about NLB by executing the **wlbs params** command.

2 Discovery Workflow

In the Discovery Control Panel window, activate the following jobs:

- **Host Connection by Shell (Discovery Modules > Network Discovery – Basic).** Discovers Windows machines that act as the triggers for the NLB discovery.
- **MS NLB by NTCMD (Discovery Modules > Cluster and Load Balancing Solutions – Microsoft NLB).** Connects to the host by NTCmd and retrieves the MS NLB Cluster topology.

For details on the discovery mechanism, see "Discovery Mechanism" on page 52.

To view the CIs that are discovered, see the Statistics table in the Details tab.

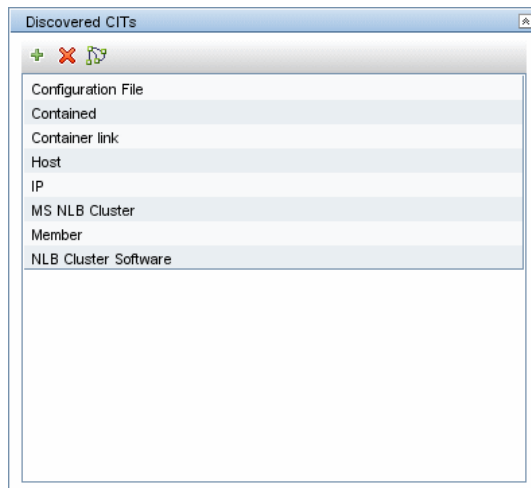
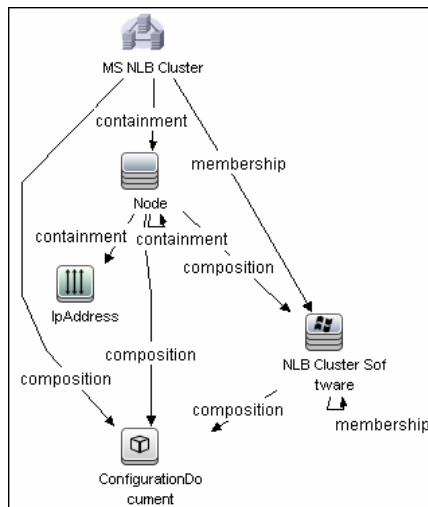
3 Packages

All components responsible for the Microsoft NLB cluster are bundled in the **Microsoft_NLB_Cluster** package (Application category).

For details, see "Package Manager" in *HP UCMDB Administration Guide*.

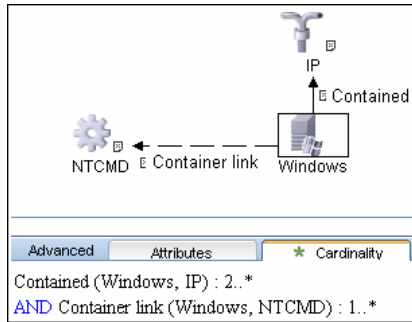
4 Discovered CITs

DFM discovers the following CITs:



- MS NLB Cluster. For details, see "MS NLB Cluster CIT" on page 59.
- NLB Cluster Software. For details, see "NLB Cluster Software CIT" on page 61.
- Configuration File. For details, see "Configuration File (NLB Port Rule)" on page 62.

5 Trigger TQL



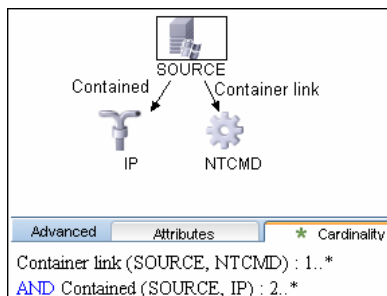
Condition. NTCMD running on a Windows machine with at least two IP addresses.

Name	Category	Description
ntcmd_with_2_IP	Trigger	Used by the MS NLB by NTCMD job
MS NLB topology	View	Used by the MS NLB Topology view

6 MS NLB by NTCMD Adapter

Trigger CIT. NTCMD

Input TQL. NTCMD running on a Windows machine with at least two IP addresses:



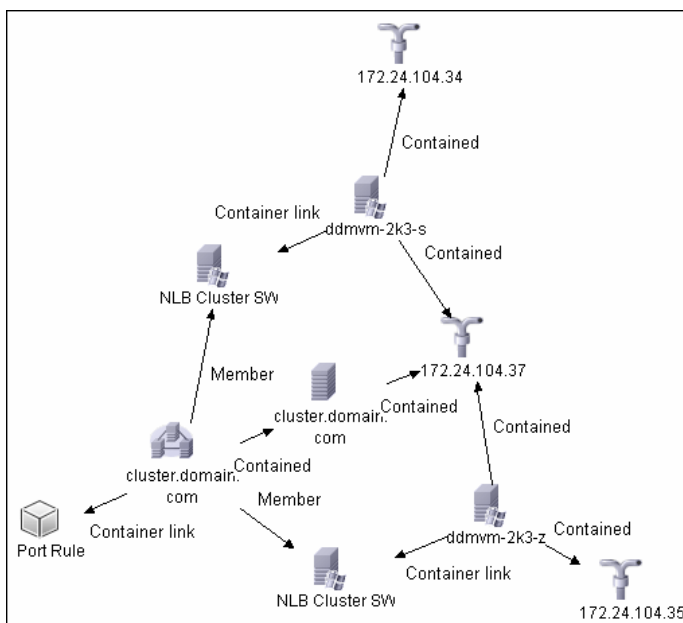
Triggered CI Data.

Name	
credentialsId	\${NTCMD.credentials_id}

7 Views

- Microsoft NLB topology

8 Topology



Reference

MS NLB Cluster CIT

The CIT represents information regarding the NLB cluster.

CIT name. ms_nlb_cluster

Parent CIT name. loadbalancecluster

Links

Start Node	Start Node Cardinality	Link Name	End Node	End Node Cardinality
ms_nlb_cluster	1..*	member	nlb_clustersoftware	1..*

The Cluster IP address is a key field, as this is the most reliable way of discovering NLB. By comparison, discovering NLB through the Cluster network address is less reliable as it is dependent on the IP address and the operating mode—Unicast, Multicast, or IGMP. The Cluster domain name is retrieved for the Cluster name.

Attributes

The following attributes are specific to the MS NLB Cluster CIT:

Key	Display Name	Attribute Name	Type
X	ClusterIPAddress	cluster_ip_address	String(15)
	ClusterNetworkMask	cluster_network_mask	String(15)
	McastIPAddress	mcast_ip_address	String(15)
	ClusterDomainName	cluster_domain_name	String(256)
	ClusterNetworkAddress	cluster_network_address	MAC Address
	IPToMACEnable	ip_to_mac_enable	Boolean
	MulticastSupportEnable	multicast_support_enable	Boolean
	IGMPSupport	igmp_support	Boolean
	RemoteControlEnabled	remote_control_enabled	Boolean
X	data_name	data_name	String (modified for this CIT)

NLB Cluster Software CIT

The CIT represents information regarding a single machine configuration that is part of an NLB cluster.

CIT name: nlb_clustersoftware

Parent CIT name. failoverclustersoftware

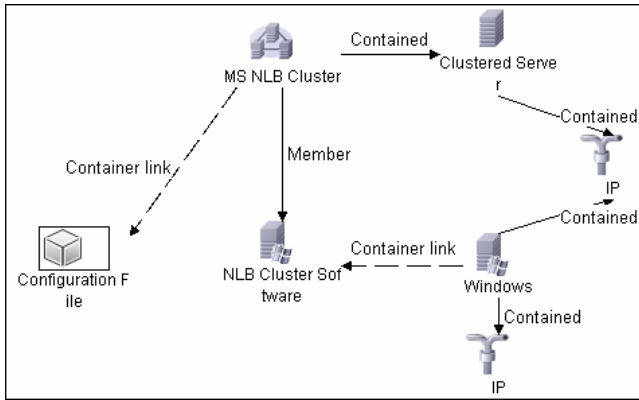
Links

Start Node	Start Node Cardinality	Link Name	End Node	End Node Cardinality
ms_nlb_cluster	1..*	member	nlb_clustersoftware	1..*
nt	1..*	container_f	nlb_clustersoftware	1..*

Attributes

Key	Display Name	Type
X	ClusterIPAddress	String(15)
	HostPriority	int (1-32)
	ClusterModeOnStart	Started, Suspended, Stopped
X	Name	String (NLB Cluster SW)
X	Container	Link

Configuration File (NLB Port Rule)



This CIT retrieves information about each port rule defined for NLB clusters.

Since the Port Rule entity cannot clearly define key attributes, the port rules properties are stored in the properties file (key=value pairs) as follows:

```
portRule1.ServingIP=All
portRule1.StartPort=0
portRule1.EndPort=100
portRule1.Protocol=Both
portRule1.FilteringMode=Multiple
portRule1.Affinity=Single
portRule1.LoadWeight=40
```

Links

Start Node	Start Node Cardinality	Link Name	End Node	End Node Cardinality
nt	1..*	container_f	nlb_clustersoftware	1..*
ms_nlb_cluster	1..*	member	nlb_clustersoftware	1..*

 **Glossary****Cluster**

A group of independent computers that work together to run a common set of applications and provide the image of a single system to the client and application. The computers are physically connected by cables and programmatically connected by cluster software. These connections allow computers to use problem-solving features such as failover in Server clusters and load balancing in Network Load Balancing (NLB) clusters. For details, refer to [http://technet.microsoft.com/en-us/library/cc784941\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc784941(WS.10).aspx).

NLB Node

Machine-participant of an NLB cluster. For details, refer to [http://technet.microsoft.com/en-us/library/cc758834\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc758834(WS.10).aspx).

Operating Mode

The NLB cluster has two operating modes:

- ▶ In its default unicast mode of operation, NLB reassigns the station (MAC) address of the network adapter for which it is enabled and all cluster hosts are assigned the same MAC (media access control) address.
- ▶ In multicast mode, NLB assigns a layer 2 multicast address to the cluster adapter instead of changing the adapter's station address. For details, refer to [http://technet.microsoft.com/en-us/library/cc783135\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc783135(WS.10).aspx).

Port Rules

The NLB driver uses port rules that describe which traffic to load-balance and which traffic to ignore. By default, the NLB driver configures all ports for load balancing. You can modify the configuration of the NLB driver that determines how incoming network traffic is load-balanced on a per-port basis by creating port rules for each group of ports or individual ports as required. Each port rule configures load balancing for client requests that use the port or ports covered by the port range parameter. How you load-balance your applications is mostly defined by how you add or modify port rules, which you create on each host for any particular port range.

Dedicated IP Address

The IP address of a NLB host used for network traffic that is not associated with the NLB cluster (for example, Telnet access to a specific host within the cluster). This IP address is used to individually address each host in the cluster and therefore is unique for each host.

Virtual IP Address

An IP address that is shared among the hosts of a NLB cluster. A NLB cluster may also use multiple virtual IP addresses, for example, in a cluster of multihomed Web servers. For details, refer to [http://technet.microsoft.com/en-us/library/cc756878\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc756878(WS.10).aspx).

Components of the Network Load Balancing Architecture

Component	Description
Nlb.exe	The Network Load Balancing control program. You use Nlb.exe from the command line to start, stop, and administer Network Load Balancing, as well as to enable and disable ports and to query cluster status.
Nlbmgr.exe	The Network Load Balancing Manager control program. Use this command to start Network Load Balancing Manager.
Wlbs.exe	The former Network Load Balancing control program. This has been replaced by Nlb.exe . However, you can still use Wlbs.exe rather than Nlb.exe if necessary, for example, if you have existing scripts that reference Wlbs.exe .
Wlbsprov.dll	The Network Load Balancing WMI provider.
Nlbmprov.dll	The Network Load Balancing Manager WMI provider.
Wlbsctrl.dll	The Network Load Balancing API DLL.
Wlbs.sys	The Network Load Balancing device driver. Wlbs.sys is loaded onto each host in the cluster and includes the statistical mapping algorithm that the cluster hosts collectively use to determine which host handles each incoming request.

6

Veritas

This chapter includes:

Tasks

- Discover Veritas Cluster Servers on page 67

Tasks

Discover Veritas Cluster Servers

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.

This task includes the following steps:

- "Overview" on page 68
- "Network and Protocols" on page 68
- "Discovery Workflow" on page 68
- "Discovered CITs" on page 69
- "Topology Map" on page 70

1 Overview

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.

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 component fails, another server takes over to provide the necessary service.

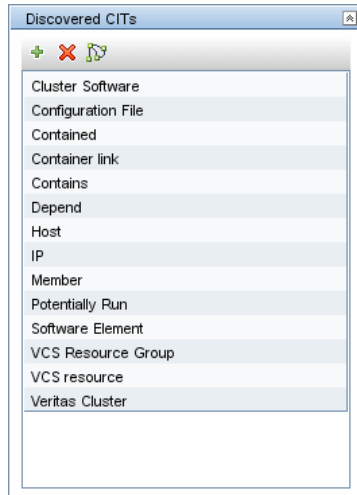
2 Network and Protocols

- ▶ **SSH.** For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.

3 Discovery Workflow

In the Discovery Control Panel window, activate the **Veritas Cluster by Shell** job.

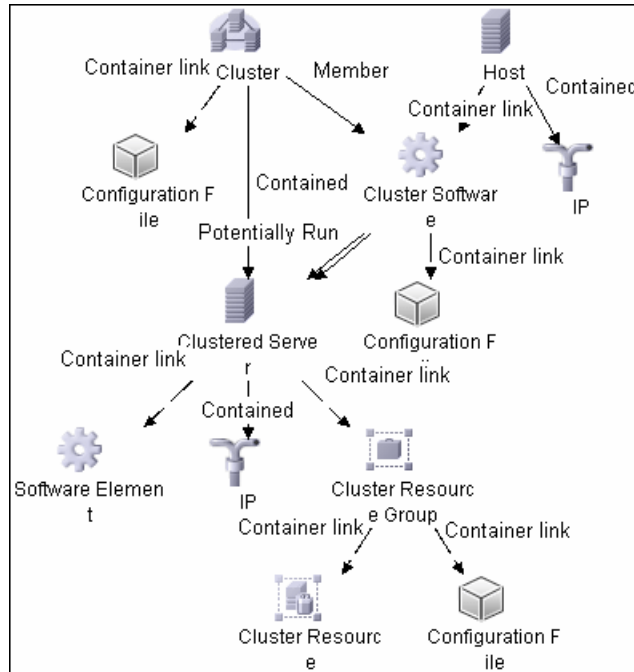
4 Discovered CITs



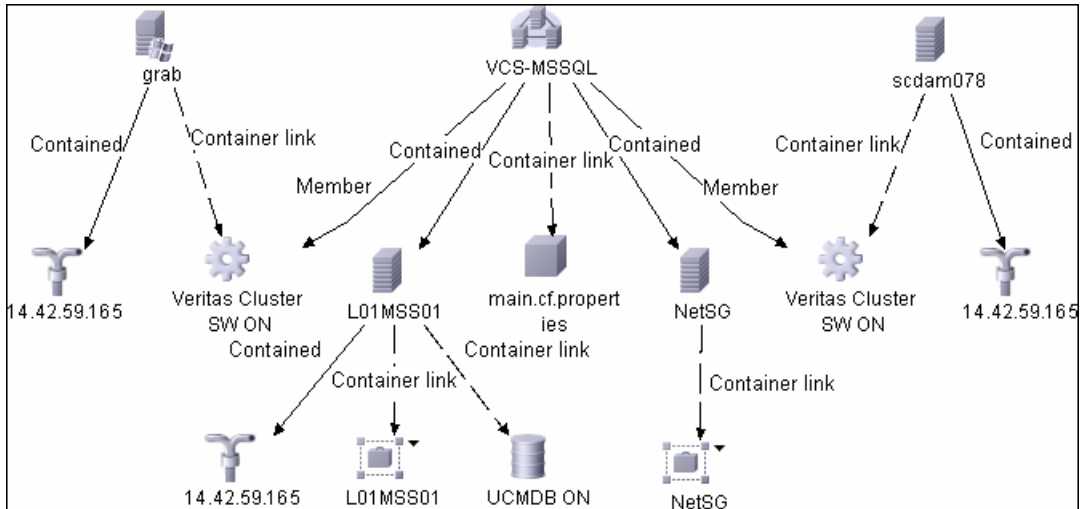
For details on the CIs that are discovered, see the Statistics Results table in the Details tab. For details, see "Statistics Results Pane" in *HP Universal CMDB Data Flow Management Guide*.

5 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.



Double-click the required node to drill down to the CIs folded underneath.



7

DB2

This chapter includes:

Tasks

- Discover IBM DB2 Databases on page 73

Tasks

Discover IBM DB2 Databases

This module discovers IBM DB2 Server databases and their components on the network.

This task includes the following steps:

- "Prerequisites" on page 73
- "Network and Protocols" on page 74
- "Discovery Workflow" on page 74
- "Discovered CITs" on page 74
- "Topology Map" on page 74
- "Troubleshooting and Limitations" on page 75

1 Prerequisites

Verify the user name, password, and port used by IBM DB2 Server.

2 Network and Protocols

IBM DB2 Server uses the **SQL protocol**. For credentials information, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*. In the Database Type box, choose **db2**.

3 Discovery Workflow

In the Discovery Control Panel window, activate the jobs in the **Database – DB2** module in the following order:

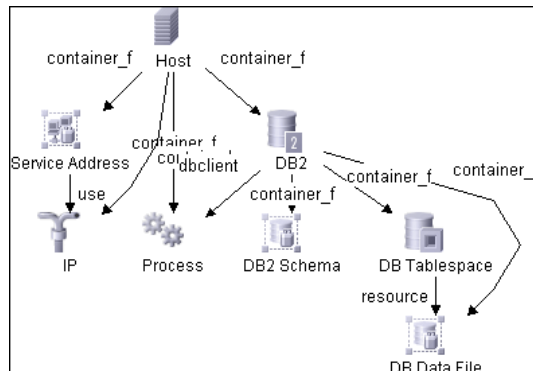
- ▶ DB2 Connection by SQL
- ▶ DB2 Topology by SQL
- ▶ Databases TCP Ports

4 Discovered CITs

For details on the CIs that are discovered, see the Statistics table in the Details tab. For details, see "Statistics Results Pane" in *HP Universal CMDB Data Flow Management Guide*.

5 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.

6 Troubleshooting and Limitations

To perform an IBM DB2 discovery, copy the following files from the installation folder on the IBM DB2 machine to the Data Flow Probe machine:

- **db2java.zip**
- **db2jcc.jar**
- **db2jcc_license_cisuz.jar**
- **db2jcc_license.jar**

Place the files in the following folder:

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\db\db2.

Restart the Data Flow Probe.

8

MS-SQL

This chapter includes:

Concepts

- ▶ Discovery by OS Credentials on page 77

Tasks

- ▶ Discover Microsoft SQL Server Database Application on page 78
- ▶ Discover SQL Server by OS Credentials on page 80

Concepts

Discovery by OS Credentials

There are two approaches to identifying MS SQL Server instance names by OS credentials. The changes appear in the **Host_Resources_Basic** package:

- ▶ **By Process Command Line.** The SQL Server process usually includes the MS SQL Server instance name in its command line. DFM extracts this instance name to a CI.

Note: A process command line cannot be retrieved by the SNMP protocol. Therefore, SNMP cannot be used to discover the MS SQL Server instance name, and DFM reports the generic running software CI instead.

- ▶ **Using Windows Services.** DFM checks existing services for those that include `sqlservr.exe` in the command line and extracts the instance name from the service name (since the service name reflects the instance name).

Tasks

Discover Microsoft SQL Server Database Application

This task describes how to discover the Microsoft SQL Server database application.

This task includes the following steps:

- ▶ "Supported Versions" on page 78
- ▶ "Prerequisites" on page 78
- ▶ "Network and Protocols" on page 78
- ▶ "Discovery Workflow" on page 79
- ▶ "Discovered CITs" on page 79
- ▶ "Topology Map" on page 80

1 Supported Versions

Microsoft SQL Server 2000, 2005, 2008.

2 Prerequisites

Verify the user name, password, and port used by Microsoft SQL Server.

3 Network and Protocols

Microsoft SQL Server uses the **SQL protocol**. For credentials information, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

In the Discovery Control Panel 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

5 Discovered CITs

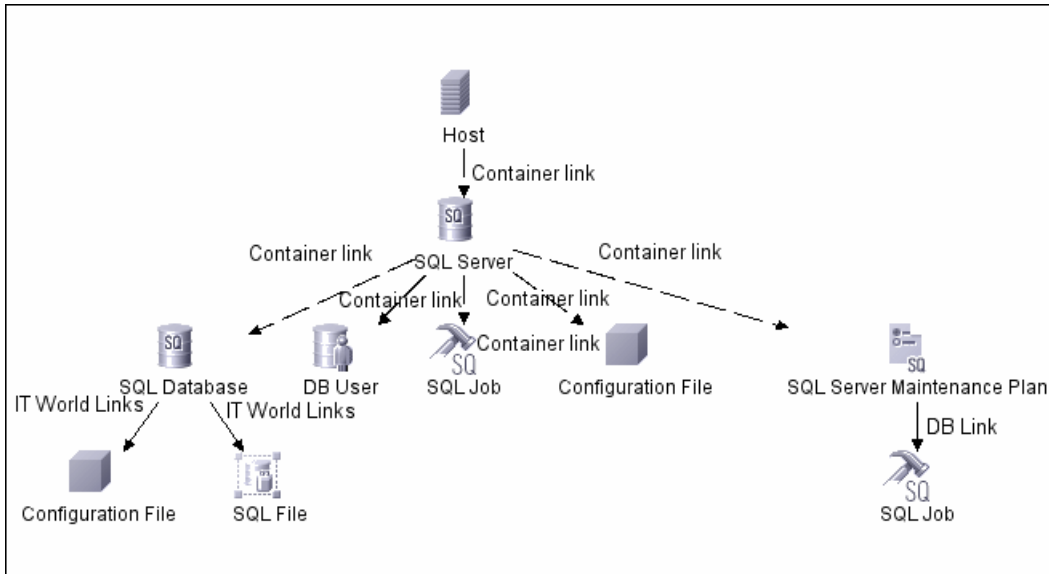
For details on the CIs that are discovered, see the Statistics table in the Details tab. For details, see "Statistics Results Pane" in *HP Universal CMDB Data Flow Management Guide*.

6 Adapter Parameters



comprehensiveDiscovery: False (the default): the SQL File, SQL Job, and DB User entities for MS SQL Server are not retrieved.

7 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.

Discover SQL Server by OS Credentials

Note: This functionality is available as part of Content Pack 3.00 or later.

This task includes the following steps:

- "Overview" on page 81
- "Discovery Jobs" on page 81
- "Discovery Changes" on page 81
- "Discovery When Host Information Is Available" on page 82
- "Discovery When Host Information Is Not Available" on page 82

1 Overview

This section describes how DFM discovers MS SQL Server CIs, using operating system (OS) credentials. DFM creates an identifiable SQL Server CI, rather than a generic RunningSoftware CI.

Previously, SQL Server discovery assumed the existence of a process with the name of **sqlservr.exe**. Once DFM found this process, generic running software with a **MSSQL DB** value in the **data_name** attribute was reported to UCMDB.

The following issues are rectified:

- ▶ DFM did not discover instance name information, so identifiable CIs could not be reported.
- ▶ When several instances of SQL Server existed in the environment, DFM reported only one running software to UCMDB.

2 Discovery Jobs

The following jobs discover MS SQL Server components by OS credentials:

- ▶ Host Resources and Applications by Shell
- ▶ Host Resources and Applications by WMI

3 Discovery Changes

The following changes have been made:

- ▶ When running discovery with OS credentials, an identifiable SQL Server CI is reported, instead of the generic RunningSoftware CI.
- ▶ DFM can now resolve the SQL Server instance name.
- ▶ The Data Flow Probe can report multiple SQL Server instances, each of them linked by a **depend** link to its own **sqlservr.exe** process.
- ▶ DFM now supports SQL Server named instances.
- ▶ DFM supports discovery of the following SQL Server versions: MSSQL 2000, 2005, 2008.

4 Discovery When Host Information Is Available

DFM runs the following SQL command:

```
select SERVERPROPERTY ('InstanceName')
```

5 Discovery When Host Information Is Not Available

DFM runs the following SQL command:

```
select @@servername
```

9

MySQL Replication Between Databases

Note: This functionality is available as part of Content Pack 4.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 83

Tasks

- ▶ Discover MySQL Configuration and Replication Jobs on page 84

Concepts

Overview

This chapter explains how to discover MySQL database servers that replicate data in a master-slave relationship.

Replication enables data from one MySQL database server (the master) to be replicated to one or more MySQL database servers (the slaves). For details on replication, see the MySQL manual on the MySQL Web site:
<http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html>.

Currently all information about databases is retrieved through Shell protocols from the MySQL configuration file.

The job responsible for MySQL discovery is **MySQL by Shell** (Database – MySQL module).

Tasks

Discover MySQL Configuration and Replication Jobs

This task describes how to discover the MySQL configuration and replication jobs.

This task includes the following steps:

- "Supported Versions" on page 85
- "Prerequisites – User Permissions" on page 85
- "Required Protocols" on page 85
- "Discovery Workflow" on page 85
- "The MySQL by Shell Job" on page 86
- "Trigger TQL" on page 87
- "Configuration Item Types" on page 87
- "CIT Attributes" on page 87
- "Links" on page 88
- "Discovered CITs" on page 89
- "The MySQL Package" on page 89
- "Input TQL" on page 90
- "Triggered CI Data" on page 90
- "Views – MySQL Replication Topology" on page 90
- "Limitation" on page 91

1 Supported Versions

- ▶ MySQL versions 4.x and 5.x are supported.
- ▶ The following operating systems are supported: Windows, Solaris, and Linux.

2 Prerequisites – User Permissions

To retrieve all relevant information, DFM must have read permissions for the \$MYSQL_HOME directory and for executing **mysqld** (**mysqld.exe** or **mysqld-nt.exe** for Windows) with the following parameters:

```
mysqld --verbose --help
```

```
mysqld --version
```

If the **my.cnf** (**my.ini**) file is located outside the \$MYSQL_HOME directory, you must add permissions for reading to it.

3 Required Protocols

Verify that the following protocols are configured:

- ▶ **SSH**. For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet**. For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **NTCmd**. For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

- a** Run the **Host Connection by Shell** job to create Shell CITs.
- b** Run any of host resources jobs to gather information about processes running on the host.
- c** Run the **MySQL by Shell** job to retrieve information about MySQL configuration and replication jobs. For details, see the following step.

5 The MySQL by Shell Job

This section explains how DFM discovers the MySQL server:

- ▶ The MySQL by Shell job connects to the remote host using Shell credentials.
- ▶ The job checks for the existence of the path of the MySQL configuration file by executing the following command:

```
mysqld --verbose --help
```

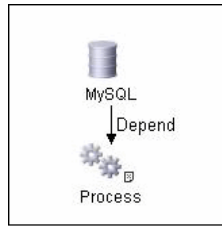
- ▶ If the job cannot find the configuration file with this command, it assumes the file is located in the default configuration file path:
 - ▶ UNIX or Linux: **/etc/my.cnf**
 - ▶ Windows: **../my.ini**
- ▶ The job tries to retrieve the attribute values from the configuration file. The job either reads the attribute values from the command line, or reads the configuration file to find the values of the attributes that were not found in the command line.

Example of command line with attribute values:

```
mysqld-nt.exe --defaults-file=C:\hp\UCMDB\DataFlowProbe\MySQL\my.ini  
DDM_Probe_DB
```

- ▶ If the job does not find any attribute values, it takes the default values from the MySQL documentation.
For details of the MySQL attributes, see "CIT Attributes" on page 87.
- ▶ The job creates the MySQL CIs with appropriate attribute values and links.
- ▶ The job now checks if this MySQL instance is a replica. If it is a replica, the job attempts to discover a master host and master user. The version of the MySQL engine is taken from the **mysqld --version** command output.
- ▶ The job creates the MySQL replication CI with appropriate attribute values and links.

6 Trigger TQL



7 Configuration Item Types

Name	Parent CIT	Uses Existing Attributes	Uses New Attributes	Description
MySQL	Database	database_dbsid	server_id, database_datadir, database_max_ connections	CIT represents the MySQL database
MySQL Replication	DB Scheduler Job		master_user, master_connect_ retry	CIT represents the MySQL Replication job

8 CIT Attributes

MySQL

- **server_id**. The unique ID for each slave and master node.
- **database_datadir**. Path to the database root (**datadir** in the configuration file).
- **database_max_connections**. The maximum number of concurrent sessions allowed by the MySQL server (**max_connections** in the **my.ini** file).
- **database_dbsid**. The unique identifier for running the MySQL instance-process port. The format is **MySQL on port #####**.

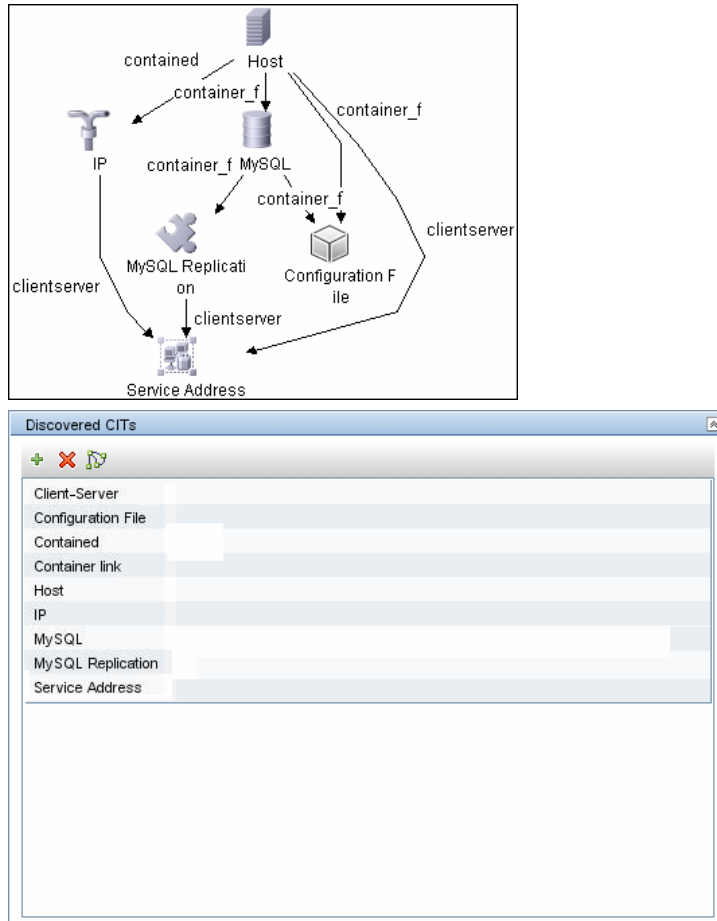
MySQL Replication

- **master_user.** A user name used when connecting to the master server.
- **master_connect_retry.** The number of seconds that the slave thread sleeps before trying to reconnect to the master, if the master goes down or the connection is lost.

9 Links

Source	Destination	Link Type	Cardinality
mysql	configfile	Container link	1..1
mysql	mysql_replication	Container link	1..1
mysql_replication	service_address	Client-Server	1..1

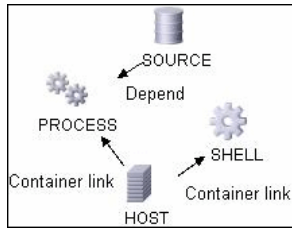
10 Discovered CITs



11 The MySQL Package

All components responsible for MySQL discovery by Shell in DFM are bundled in the MySQL package (in the Database category in the Package Manager). For details, see "Package Manager" in the *HP UCMDB Administration Guide*.

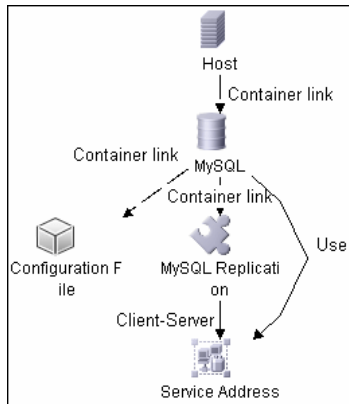
12 Input TQL



13 Triggered CI Data

Triggered CI Data	
Name	
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
dbport	\${SOURCE.database_dbport}
dbsid	\${SOURCE.database_dbsid}
ip_address	\${SHELL.application_ip}
processParams	\${PROCESS.process_parameters}
processPath	\${PROCESS.process_path}

14 Views – MySQL Replication Topology



15 Limitation

There are two main approaches to running several active MySQL instances on one host:

- ▶ Two MySQL instances are each run on a different port, for example, one on 134.44.1.1:3306 and the second on 134.44.1.1:3307.
- ▶ A host has several IPs, and each MySQL process is bound to its own IP, for example, 134.44.1.1:3306 and 134.44.1.2:3306.

In the second case, as the key identifier that differentiates one MySQL CI from another is a port number (without an IP), the job cannot differentiate between the two MySQL instances and merges them into one CI.

10

Oracle

This chapter includes:

Tasks

- Discover Oracle Databases on page 93
- Discover Oracle Real Application Cluster (RAC) on page 95

Tasks

Discover Oracle Databases

This task describes how to discover Oracle databases. This discovery adds a valid credentials ID to the CMDB. You can then use this CI to fully discover the database.

This task includes the following steps:

- "Supported Versions" on page 93
- "Prerequisites" on page 94
- "Network and Protocols" on page 94
- "Discovered CITs" on page 94
- "Topology Map" on page 95

1 Supported Versions

Oracle 8, 9, 10.

2 Prerequisites

Run **Databases TCP Ports**. (The ports that are discovered are those open TCP\UDP ports running on a host with known server ports.)

3 Network and Protocols

To discover Oracle databases, use the following protocol:

SQL. For credentials information, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*.

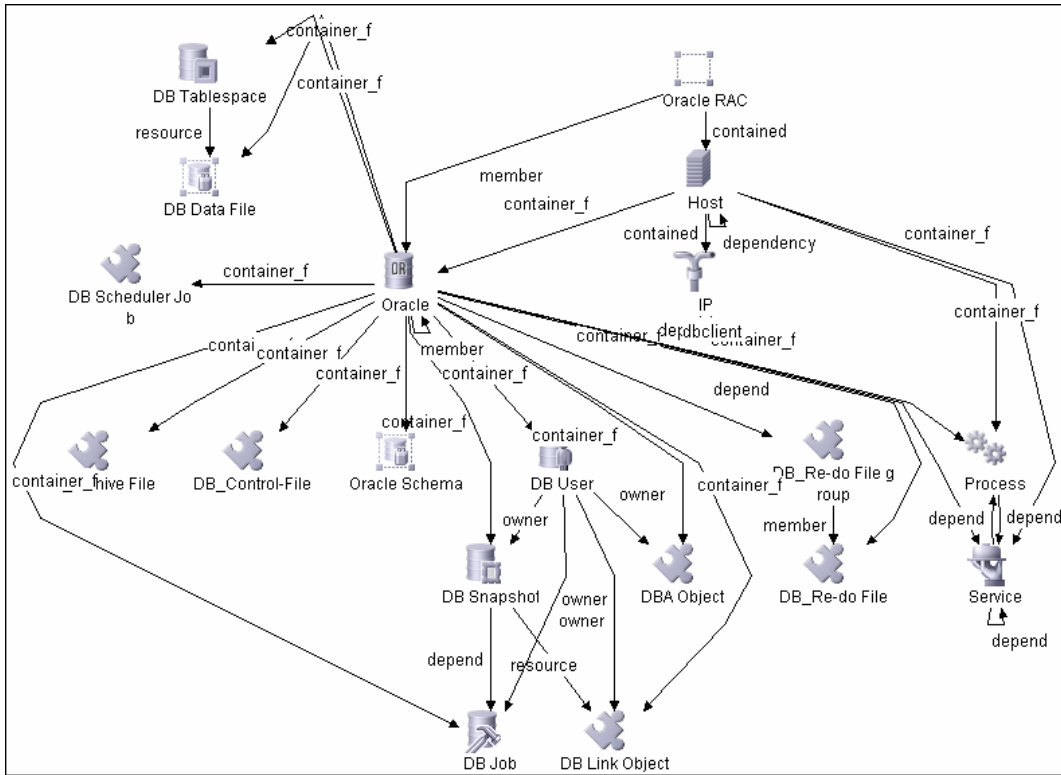
4 Discovered CITS

owner, dbjob, dbuser, process, dbclient, dblinkobj, dbsnapshot, dbdatafile, dbtablespace, db_controlfile, db_redofile, db_redofilegroup, db_archivefile, oracle, dbschedulerjob, service, rac

The following attributes are updated:

- database_dbversion
- database_dbtype
- database_dbsid
- database_dbport

5 Topology Map



Discover Oracle Real Application Cluster (RAC)

Note: This functionality is available as part of Content Pack 4.00 or later.

This section explains how to run the **Oracles Listeners by Shell** and the **Oracle RAC Topology by Shell** jobs.

This section includes the following topics:

- "Overview" on page 96
- "Supported Versions" on page 96
- "Prerequisites" on page 97
- "Required Protocols" on page 97
- "Discovery Workflow" on page 97
- "The Oracle Listeners by Shell Job" on page 98
- "The Oracle RAC Topology by Shell Job" on page 100
- "Topology" on page 103
- "Configuration Items" on page 103
- "The Oracle Package" on page 104
- "Oracle View" on page 105
- "Troubleshooting and Limitations" on page 105

1 Overview

DFM discovers information about Oracle RAC through the Shell protocols from the Oracle configuration files **listener.ora** and **tnsnames.ora**, and through the **lsnrct** utility.

The following jobs are responsible for Oracle RAC discovery (in the **Database – Oracle** module):

- Oracle Listeners by Shell
- Oracle RAC Topology by Shell

2 Supported Versions

Oracle RAC over Oracle DB 10 and 11 is supported.

3 Prerequisites

- a** To retrieve all relevant information, verify that DFM has:
 - Read permissions for the `$ORACLE_HOME\network\admin` directory
 - The correct execute permissions for `$ORACLE_HOME\bin\lsnrctl` and for the corresponding library (lib) and message files.
- b** **The Oracle Listeners by Shell job.** Verify that the RAC relative processes are running on the Oracle database. The file names begin with `ora_lms`, `ora_lmd`, `ora_lck`, and `oracm`.
- c** **The Oracle RAC Topology by Shell job.** The **Listened IPs** of the Listener CIT must be **not NULL**.
- d** Run the **Host Connection by Shell job**, to activate Shell CITs.

4 Required Protocols

Verify that the following protocols are configured:

- **NTCmd.** For credentials information, see "NTCMD Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- **SSH.** For credentials information, see "SSH Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- **Telnet.** For credentials information, see "Telnet Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

5 Discovery Workflow

- a** Run any of the host resources jobs that gather information about processes running on the host.

If DFM discovers TNS Listener processes, the job creates Oracle TNS Listener CIs and an Oracle DB CI together with its connected processes.
- b** To discover Oracle TNS Listener CIs with full data, run the **Oracle Listeners by Shell job**. This job connects to the host and retrieves the required data for the Oracle TNS Listener CI. For details, see "The Oracle Listeners by Shell Job" on page 98.

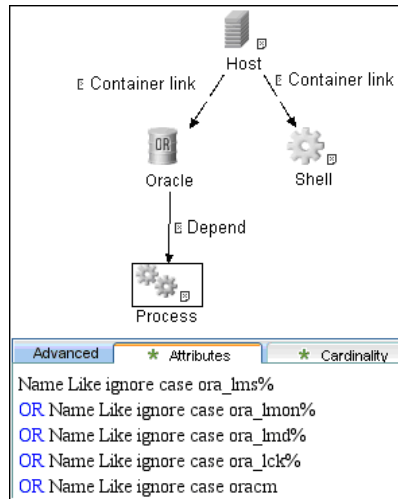
- c To discover Oracle RAC topology, run the **Oracle RAC Topology by Shell** job. This job connects to the hosts with full listeners and discovers RAC. For details, see "The Oracle RAC Topology by Shell Job" on page 100. For details on undiscovered elements, see "Troubleshooting and Limitations" on page 105.

6 The Oracle Listeners by Shell Job

This job triggers on Oracle databases that have RAC related processes. The job:

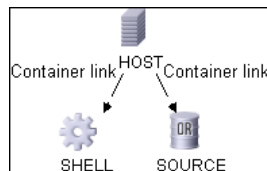
- Connects to the remote host by Shell.
- Checks for the **ORACLE_HOME** environment variable.
If the variable is not defined, the job takes the **ORACLE_HOME** value from the job adapter (if defined).
- Reads the **Oracle TNS listener** configuration file, stored in **\$ORACLE_HOME/network/admin/listener.ora**, and performs further parsing.
- Retrieves a full list of IP addresses to which this particular listener is listening.
- Checks for listener status using the **\$ORACLE_HOME/bin/lsnrctl** status.
- Retrieves known services and listener status from the output.

Trigger TQL



The adapter used by the job is **Oracle_Listeners_by_Shell**. To access:
Adapter Management > Discovery Resources pane > Oracle > Adapter > Oracle_Listeners_by_Shell.

Input TQL



► **Used Scripts.** oracle_listeners_by_shell.py

Triggered CI Data

Triggered CI Data	
+	✕
Name	
credentialsId	\${SHELL.credentials_id}
ip_address	\${SHELL.application_ip}

Discovered CITs



Discovery Adapter Parameters

- ▶ **OracleHomes.** Used when no **ORACLE_HOME** environment variable is defined. This value must be the same as the parameter in the Oracle RAC Topology by Shell job.

7 The Oracle RAC Topology by Shell Job

This job:

- ▶ Connects to the remote host by Shell.
- ▶ Checks for the **ORACLE_HOME** environment variable.
If it is not defined, the job uses the **OracleHome** value from the job adapter.
- ▶ Retrieves RAC parameters such as Service Name and Nodes from the **\$ORACLE_HOME/network/admin/tnsnames.ora** file.
- ▶ Checks if this RAC instance is running, by parsing the **lsnrctl status** output.

Note: Nodes are cited in the **tnsnames.ora** file by their internal IP or by their internal domain name. If the domain name appears, DFM resolves it.

- ▶ Retrieves the full list of Listened IPs from the input TQL, for all listeners matching the TQL.

- Parses this attribute's values from the list of listened IPs, to retrieve the Host Primary Domain name that corresponds to the MAC address.

This is needed since the RAC CI's **data_name** key attribute must consist of a list of all the node domain names separated by the colon symbol (:).

- Looks up the full node name in the build table sorted by IP address.

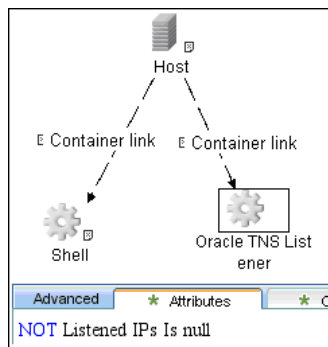
The result is the Host Primary Domain name for each node.

At this stage, the following information is available: the RAC Service Name, the fully qualified domain names of all the RAC nodes, and a RAC instances count.

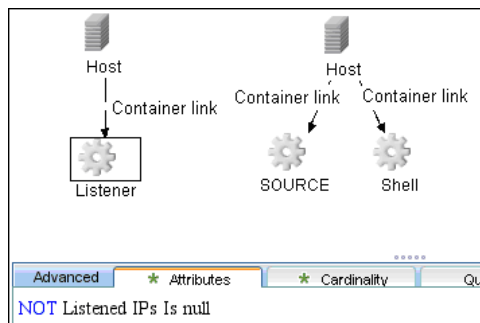
- Creates the RAC CI.

The adapter used by the job is **Oracle_RAC_Topology_by_Shell**.

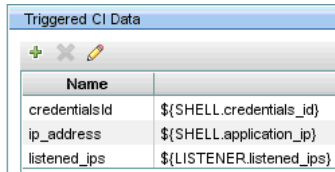
Trigger TQL



Input TQL

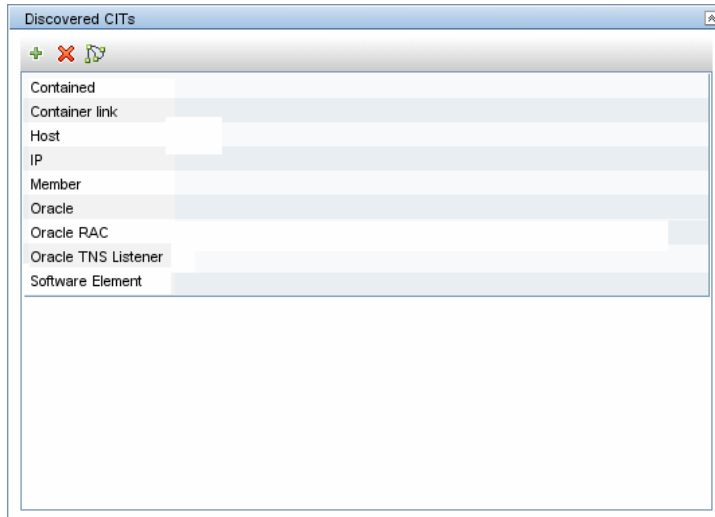


Triggered CI Data



Name	
credentialsId	\${SHELL.credentials_id}
ip_address	\${SHELL.application_ip}
listened_ips	\${LISTENER.listened_ips}

Discovered CITs

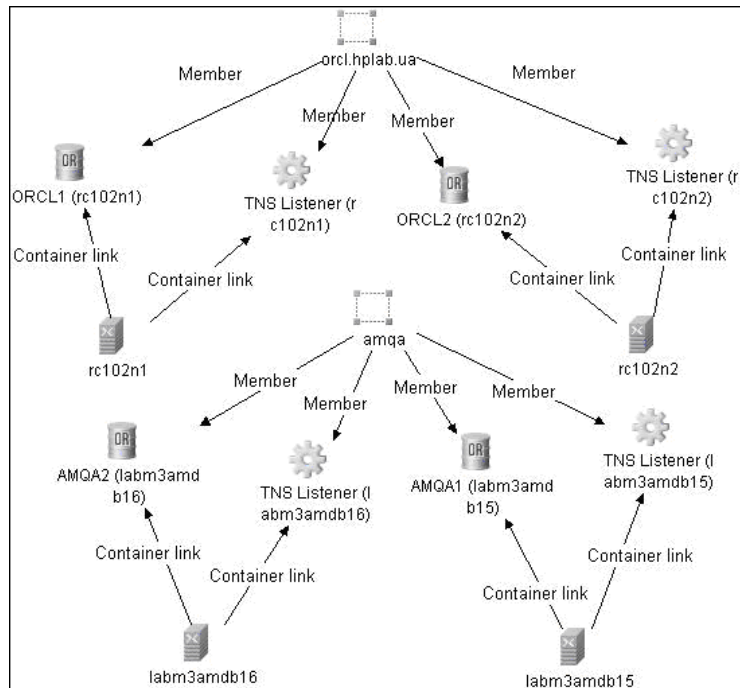


Contained
Container link
Host
IP
Member
Oracle
Oracle RAC
Oracle TNS Listener
Software Element

Discovery Adapter Parameters

OracleHomes. Used when no **ORACLE_HOME** environment variable is defined. This value must be the same as the parameter in the Oracle Listeners by Shell Job job.

8 Topology



9 Configuration Items

- **Oracle TNS Listener.** This CIT represents the Oracle TNS Listener.
- **CIT name.** oracle_listener
- **Parent CIT name.** application
- **Key attributes:**
 - **data_name (displayed as Name).** The TNS Listener constant.
 - **root_container (displayed as Container).** The Container CI.
 - **listener_name (displayed as Name of the Listener).** The real TNS Listener name.

Additional Attributes

listened_ips (displayed as Listened IPs). Listened to IP addresses and machine domain name. Listened IPs are IP addresses that are listened to by the Oracle TNS Listener.

Format:

```
<host_name>:<host_primary_ip>@<listened_ip>:<mac>;...
<listened_ip>:<mac>
```

Note: MAC addresses are not currently discovered. The marker acts as a placeholder for future enhancements.

Links

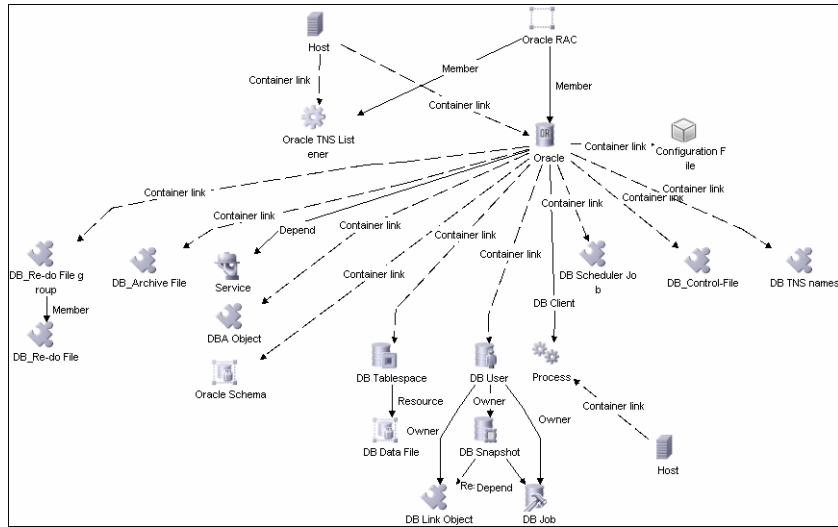
CIT	Link Type	Cardinality
Host	Container link	1.*
RAC	Member	1.*
Process	Depend	1.*

10 The Oracle Package

All components responsible for Oracle RAC discovery are bundled in the **Oracle** package (**Administration > Package Manager > Oracle**), under the Database category.

11 Oracle View

To access: **Modeling > View Manager > Views > Root > Database > Oracle > Oracle.**



12 Troubleshooting and Limitations

This section describes troubleshooting and limitations for Oracle discovery.

Error Message	Description
Failed to lookup host name. No RAC CI will be created.	<p>For one or more nodes, the job failed to retrieve the FQDN (fully qualified domain name) from the listeners listened_ips attribute information.</p> <ul style="list-style-type: none"> ▶ Check the logs to retrieve the IP and destination. ▶ Make sure that the FQDN for that IP can be obtained either from the DNS or from the host file.

Error Message	Description
No RAC CI are retrieved.	Not all nodes were discovered with the correct listener information.
Discovery cannot discover links to the remote machines (database clients)	This can occur 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.

11

Active Directory

Note: This functionality is available as part of Content Pack 5.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 107

Tasks

- ▶ Discover Active Directory Domain Controllers and Topology on page 108

Concepts

Overview

Active Directory (AD) provides an extensible and scalable directory service that enables efficient managing of network resources.

DFM discovers Active Directory topology through the LDAP Directory Service Interface that communicates with the AD domain controllers. DFM uses JNDI to provide the API that interacts with the LDAP Directory Service Interface.

Tasks

Discover Active Directory Domain Controllers and Topology

This task explains how to discover Active Directory and includes the following steps:

- "Supported Servers" on page 108
- "Prerequisites" on page 108
- "Network and Protocols" on page 109
- "Discover AD Domain Controllers" on page 110
- "Discover AD Topology" on page 111
- "Active Directory Topology" on page 114

1 Supported Servers

- Windows Server 2000
- Windows Server 2003
- Windows Server 2008

2 Prerequisites

- a** Discover the host of each AD domain controller: activate one of the following jobs (depending on the protocol you are using) in the **Network – Basic** module:
 - **Host Connection by Shell**
 - **Host Connection by SNMP**
 - **Host Connection by WMI**

- b** Verify that the **portNumberToPortName.xml** configuration file includes all possible AD ports. For example, if AD is running on LDAP port 389, locate the following row in the file:

```
<portInfo portProtocol="tcp" portNumber="389" portName="ldap" discover="0" />
```

Change the **discover="0"** attribute value to **discover="1"**.

For details, see "Define a New Port" and "The portNumberToPortName.xml File" in *HP Universal CMDB Data Flow Management Guide*.

- c** Open the LDAP port of the destination IP for each domain controller server, by activating the following job in the **Network – Advanced** module:
 - TCP Ports
 - This job includes the **TCP_NET_Dis_Port** adapter.

3 Network and Protocols

- a** To discover hosts, you must set up the SNMP, Shell (NTCMD, SSH, Telnet), and WMI protocols:
 - **SNMP**. For credentials information, see "SNMP Protocol" in *HP Universal CMDB Data Flow Management Guide*.
 - Prepare the following information for the SNMP protocol: **community name** (for v2 protocol), **user name** (for v3 protocol), and **password** (for v3 protocol).
 - **NTCMD**. For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
 - **SSH**. For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
 - **Telnet**. For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.

Prepare the following information for the Shell protocols: **user name**, **password**, and **domain name** (optional for NTCMD).

- **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*.

Prepare the following information for the WMI protocol: **user name**, **password**, and **domain name** (optional).

- b** To run all AD jobs, you must set up the LDAP protocol. There are two versions of the protocol available: **2** and **3**. As version 2 has never been standardized in any formal specification, DFM uses the version 3 protocol.

For details on configuring the LDAP protocol, see "LDAP Protocol" in *HP Universal CMDB Data Flow Management Guide*.

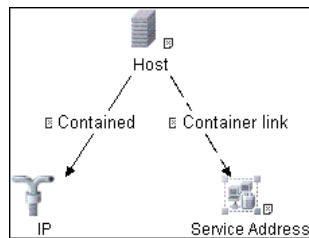
Note: User Name: if a domain is present, use **username@domain**.

4 Discover AD Domain Controllers

In the Discovery Control Panel window, activate the **Active Directory Connection by LDAP** job (in the **Application – Active Directory** module). This job discovers the existence of AD domain controllers through LDAP.

Trigger CI: IP

Trigger TQL:



CI Attributes:

CI	Attribute Value
Source	NOT IP Probe Name Is null
Service Address	Name Equal ignore case "ldap"

Triggered CI Data:

Name	Value	Description
hostId	\${HOST.root_id}	The ID of the host on which the domain controller resides.
ip_address	\${SOURCE.ip_address}	The IP address, retrieved from the service address.
port_number	\${Service_Address.ippport_number}	The LDAP port number, retrieved from the service address.

Discovered CITs:

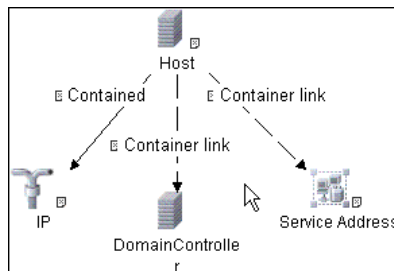
- Contained
- Container link
- DomainController
- Host
- IP

5 Discover AD Topology

In the Discovery Control Panel window, activate the **Active Directory Topology by LDAP** job (in the **Application – Active Directory** module). This job connects to the AD domain controller servers and discovers their topology.

Trigger CI: DomainController

Trigger TQL:



CI Attributes:

CI	Attribute Value
IP	NOT IP Probe Name is null
Source	<ul style="list-style-type: none"> ▶ NOT Reference to the credentials dictionary entry is null ▶ NOT Application IP is null
Service Address	Name Equal ignore case "ldap"

Triggered CI Data:

Name	Value	Description
application_port	\${SOURCE.application_port:NA}	The port retrieved from the service address.
credentialsId	\${SOURCE.credentials_id}	The credentials ID of the protocol saved in the domain controller's attribute.
hostId	\${HOST.root_id}	The ID of the host on which the domain controller resides.
ip_address	\${SOURCE.ip_address}	The IP address of the server.
port	\${SERVICE_ADDRESS.ipport_number}	The LDAP port number.

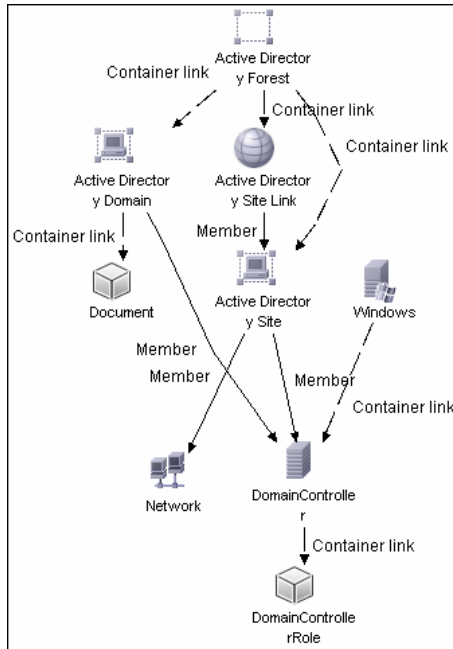
Discovery Adapter Parameters:

- ▶ **tryToDiscoverGlobalCatalog.** If this parameter is set to **true**, DFM attempts to discover the entire topology by connecting to the domain controller designated as a global catalog server. The connection is made through the port defined in the **globalCatalogPort** parameter. By default, the global catalog is used for discovery, so the default is **true**.
- ▶ **globalCatalogPort.** The port number through which DFM accesses the domain controller designated as the global catalog. The default value is **3268**. This parameter is needed only when **tryToDiscoverGlobalCatalog** is set to **true**.

Discovered CITs

- **Active Directory Domain.** Domains in the AD Forest.
- **Active Directory Forest.** Information about functionality level and contiguous names.
- **Active Directory Site.** Available site objects that are configured in the AD Forest.
- **Active Directory Site Link.**
- **Active Directory System.**
- **Container link.**
- **Document.**
- **DomainController**
- **DomainControllerRole**
- **Host.**
- **Member.** Relationships between sites and subnets.
- **Network.** Available subnet objects.

6 Active Directory Topology



12

Microsoft Exchange

This chapter includes:

Concepts

- ▶ Overview on page 115

Tasks

- ▶ Discover Microsoft Exchange Server 2003 on page 116
- ▶ Discover Microsoft Exchange Server 2007 on page 122
- ▶ Discover Microsoft Exchange Server Topology with Active Directory on page 125

Concepts

Overview

DFM discovers the following components of Microsoft Exchange Server (Exchange) software, versions 2003 and 2007: Microsoft Exchange System, Server, Administrative and Routing groups, and Public folders and Folder trees.

Currently, all information about Exchange is retrieved by the WMI protocol from the **root\MicrosoftExchangeV2** namespace.

There are two jobs responsible for Exchange discovery:

- ▶ Microsoft Exchange connection by WMI
- ▶ Microsoft Exchange topology by WMI

Tasks

Discover Microsoft Exchange Server 2003

This task explains how to discover Exchange 2003.

This task includes the following steps:

- "Supported Versions" on page 116
- "Prerequisites" on page 116
- "Network and Protocols" on page 117
- "Discovery Workflow" on page 117
- "Configuration Item Types" on page 118
- "Discovered CITs" on page 119
- "The Microsoft Exchange Server Package" on page 119
- "TQLs" on page 120
- "Views" on page 120
- "Topology Map" on page 120
- "Troubleshooting and Limitations" on page 121

1 Supported Versions

Microsoft Exchange Server 2003 is supported.

2 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 121.

3 Network and Protocols

WMI. For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*. Information about Exchange is taken from the `root\MicrosoftExchangeV2` namespace.

4 Discovery Workflow

In the Discovery Control Panel window, activate the following jobs:

- ▶ **Network – Basic** (Host Connection by WMI) to discover WMI CITs.
- ▶ Run any of the **Host Resources and Applications** 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 Exchange is running. The job creates Exchange 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,
MTADDataPath, MessageTrackingEnabled, MessageTrackingLogFileLifetime,
MessageTrackingLogFilePath, MonitoringEnabled, Type FROM Exchange_Server
```

This query returns all Exchange servers present in the Exchange organization.

- ▶ **Microsoft Exchange Topology by WMI.** The Exchange 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 Exchange is running and retrieves the complete topology. (For details on troubleshooting error messages, see "Troubleshooting and Limitations" on page 121.)

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, FQDN, Name, 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
```

5 Configuration Item Types

The following CIs are created for Exchange components:

a 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 organization.
- ▶ **Exchange Organization.** This CIT represents the top-level of the Exchange organization. For example, if an organization uses the Exchange solution, then all the Exchange components are linked to a single Exchange Organization CI.
- ▶ **Exchange Routing Group.** This CIT represents a Routing Group that exists in the Exchange organization. Routing groups supply varying network connectivity across servers, and restrict access of users in specific areas. Routing groups are deprecated in Exchange 2007. Instead Exchange 2007 relies on the Active Directory Sites configuration to connect between different Exchange Servers.

b Microsoft Exchange Server

This CIT is inherited from the RunningSoftware CIT. The CIT represents Exchange software installed on a host.

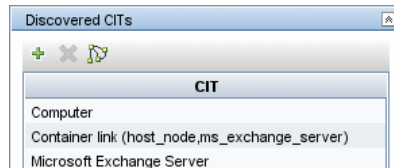
c Microsoft Exchange Resource

This CIT is located in the Application Resource folder. It is an abstract CIT that is the parent of the following CITs:

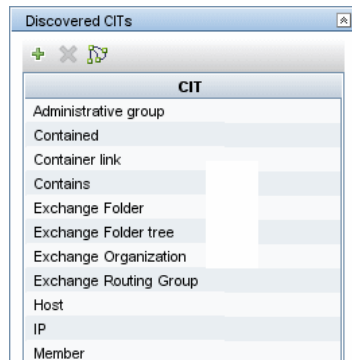
- ▶ **Exchange folder.** This CIT represents the public folders available in the Exchange organization. A public folder may be organized in a hierarchical structure, that is, one public folder may contain another public folder.
- ▶ **Exchange folder tree.** This CIT provides information about public and private folder trees on Exchange servers.

6 Discovered CITs

MS_Exchange_Connection_by_WMI:



MS_Exchange_Topology_by_WMI:



7 The Microsoft Exchange Server Package

All components responsible for Exchange in DFM are bundled in the Microsoft_Exchange_Server package.

8 TQLs

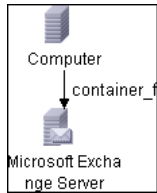
Name	Category	Used by...
ms_exchange_process_and_wmi	Trigger	Microsoft Exchange connection by WMI job
ms_exchange_server_and_host_and_wmi	Trigger	Microsoft Exchange topology by WMI job
Microsoft Exchange Topology	View	Microsoft Exchange topology view

9 Views

The following view displays Exchange components: **Microsoft Exchange topology**.

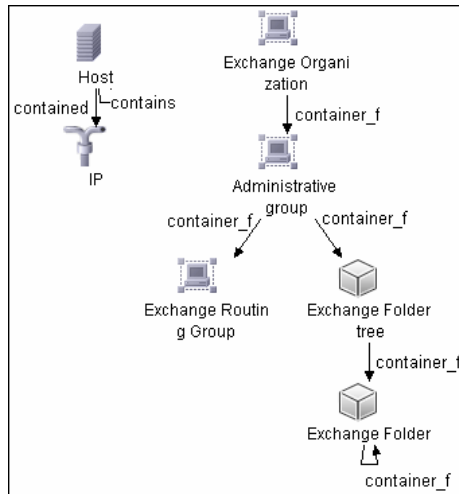
10 Topology Map

MS_Exchange_Connection_by_WMI:



MS Exchange 2003 Topology by WMI

DFM connects to the remote host and retrieves the topology for MS Exchange 2003:



11 Troubleshooting and Limitations

This section describes troubleshooting and limitations for Microsoft Exchange discovery.

- **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	<p>To model Exchange topology correctly, the Microsoft Exchange Connection by WMI job should know the name of the host to which it is connected.</p> <p>DFM tries to retrieve the host_hostname attribute of the host, matched by the input TQL. If the attribute is not set, DFM runs the following WMI query to obtain the domain name of the host:</p> <pre>SELECT Name FROM Win32_ComputerSystem</pre> <p>If this query fails for any reason, the job also fails with this error message.</p>	<ul style="list-style-type: none"> ► 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 116.

Discover Microsoft Exchange Server 2007

DFM discovers the following CIs for Microsoft Exchange Server 2007: Exchange System, Microsoft Exchange Server, and Exchange role.

DFM discovers Exchange by executing a PowerShell script on a remote machine with Exchange installed.

This task includes the following steps:

- "Prerequisites" on page 123
- "Supported Versions" on page 123
- "Network and Protocols" on page 123
- "Discovery Workflow" on page 123

- "The Microsoft Exchange Server Package" on page 124
- "Discovered CITs" on page 124
- "Topology Maps" on page 124

1 Prerequisites

- Set the script execution policy either to **Unrestricted** or **Remote Signed**.
- Verify that the account used for discovery has the permissions of the **Exchange View-Only Administrator** role.

2 Supported Versions

Microsoft Exchange Server 2007.

3 Network and Protocols

NTCmd. For credentials information, see "NTCMD Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

LDAP. For credentials information, see "LDAP Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

- a** Define NTCmd credentials. The account must have Exchange View-Only Administrator permissions.
- b** Run the **Host Connection by Shell** job.
- c** Run the **Host Resources and Applications by Shell** job to discover the Exchange process.
- d** Run the **Microsoft Exchange Connection by NTCMD** job to discover Exchange Server CIs.
- e** Run the **Microsoft Exchange Topology by NTCMD** job to discover the rest of the topology.

5 The Microsoft Exchange Server Package

All components responsible for Exchange in DFM are bundled in the **Microsoft_Exchange_Server** package. For details on the package, click the **Readme** link in the Package Manager.

6 Discovered CITs

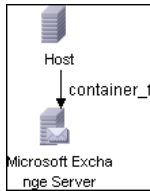
The following CITs are used to create CIs for Exchange components:

- ▶ **Exchange Organization.** This CIT represents the top-level Exchange system. For example, if an organization uses the Exchange solution, all the Exchange components are linked to a single Exchange Organization CI.
- ▶ **Microsoft Exchange Server.** This CIT is inherited from the RunningSoftware CIT. The CIT represents Exchange software installed on a host.
- ▶ **Exchange Folder.** This CIT represents Public folders available on the Exchange system. Public folder can be organized in a hierarchical structure, that is, one Public folder can contain another Public folder.
- ▶ **Exchange Message Queue.** This CIT provides properties for Microsoft Exchange queues.
- ▶ **Exchange Role.** This CIT is located in the **Application Resource > Microsoft Exchange Resource** folder. It is an abstract CIT that is the parent of the following CITs:
 - ▶ **Exchange Client Access Server.** Represents the Client Access Server role.
 - ▶ **Exchange Mail Server.** Represents the Mail Server role.
 - ▶ **Exchange Edge Server.** Represents Edge Server role.
 - ▶ **Exchange Hub Server.** Represents Hub Server role.
 - ▶ **Exchange Unified Messaging server.** Represents Unified Messaging server role.

7 Topology Maps

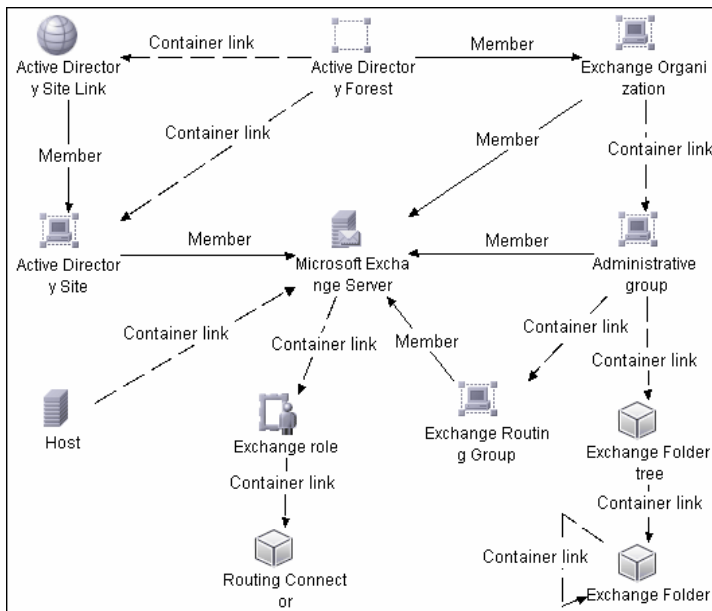
The following maps illustrate Microsoft Exchange Server 2007 topology.

MS Exchange Connection by NTCMD:



MS Exchange 2007 Topology:

DFM runs the NTCMD protocol to retrieve the topology for MS Exchange 2007:



Discover Microsoft Exchange Server Topology with Active Directory

Note: This functionality is available as part of Content Pack 5.00 or later.

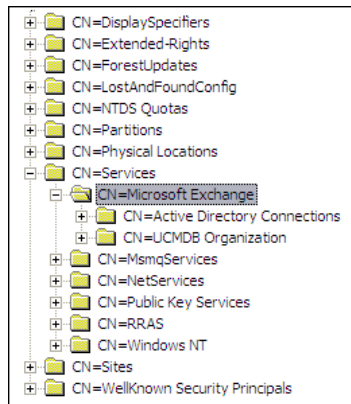
This section explains how DFM discovers Exchange by utilizing the tight integration between Exchange and AD. DFM runs jobs to discover Exchange elements in the topology that are available only through AD.

This task includes the following steps:

- "Overview" on page 127
- "Prerequisites – Set Permissions" on page 129
- "Prerequisites – Discover a Domain Controller" on page 129
- "Supported Versions" on page 130
- "Network and Protocols" on page 130
- "Discovery Workflow" on page 130
- "The Microsoft Exchange Server Package" on page 130
- "Additional CITs" on page 130
- "Deprecated CITs" on page 131
- "Modified CITs" on page 131
- "Discovered CITs" on page 131
- "Trigger TQL" on page 132
- "Trigger CI" on page 132
- "CI Attributes" on page 132
- "Adapters" on page 133
- "Topology Maps" on page 133
- "Troubleshooting and Limitations" on page 133

1 Overview

With the addition of LDAP protocol support in Content Pack 5, DFM can discover the Exchange topology using AD. Since Exchange is tightly integrated with AD and stores most of its configuration there, DFM connects to the AD Domain Controller and extracts information from it. The Exchange configuration is stored in a specific node under Services:



The Base Distinguished Name of this node is:

"CN=Microsoft Exchange, CN=Services, CN=Configuration,DC=ucmdb-ex, DC=dot"

where **ucmdb-ex.dot** is the name of the domain in this example.

If this node exists, DFM drills down and discovers all remaining information that includes: Exchange organization, Exchange servers, administrative and routing groups, connectors, roles, and so on.

Multiple Domain Controllers can serve the same domain, in which case the information is replicated between them (multi-master replication). The controllers contain the same data, so DFM needs to run only against one of them.

Note: The job for AD discovery triggers on, and runs against, all discovered domain controllers. However, as only updates are sent to the CMDB by the Data Flow Probe's result processing mechanism, the information is reported only once.

AD machines in the domain are registered in DNS as being configured for AD. DFM retrieves the FQDN (fully qualified domain name) from every Exchange discovery. This is the name of Exchange within AD. To report such an Exchange, DFM tries to resolve the FQDN to an IP address, as follows:

- ▶ DFM uses the default Data Flow Probe's DNS to resolve the Exchange FQDN.
- ▶ If this fails, DFM uses the target Domain Controller as the DNS. This is because in many cases the DNS server runs on the same machine as the Domain Controller. DFM runs the command "**netstat <FQDN> <targetDC>**" in the Data Flow Probe's local Shell.
- ▶ If this fails, DFM skips this Exchange instance.

Note: A message is displayed by the job if the FQDN cannot be resolved either by a local DNS or by using the target Domain Controller as the DNS:

Cannot resolve IP address for host '<host>', Exchange Server won't be reported

2 Prerequisites – Set Permissions

Define at least one set of LDAP protocol credential. These credentials should enable connecting to a Domain Controller through the LDAP protocol and performing searches. DFM does not modify information in AD. The queried nodes reside in the Configuration partition under the following nodes:

- **CN=Services,CN=Microsoft Exchange** node
- **CN=Sites** node

The LDAP protocol credentials should include:

- **User name and password.** Use the user account from the target domain. For all nodes that are to be queried, give **List Contents** and **Read all properties** permissions.
- **Authentication type. Simple.**

For credentials information, see "LDAP Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

3 Prerequisites – Discover a Domain Controller

To discover the Exchange topology with AD, DFM must first find a Domain Controller with an available LDAP connection.

- a** Activate the **Range IPs by ICMP** job, to ping the target host on which the Domain Controller runs (**Discovery Modules > Network Discovery > Basic**).
- b** Activate the **TCP Ports** job against the target host, to discover open LDAP ports (**Discovery Modules > Network Discovery > Advanced**).
- c** Activate the **Active Directory Connection by LDAP** job, to discover the Domain Controller on the target host (**Discovery Modules > Enterprise Applications > Active Directory**).
- d** To enable DFM to use the LDAP protocol, edit the following line in the **portNumberToPortName.xml** file (**Adapter Management > Discovery Resources > Network > Configuration Files**). Change:

```
<portInfo portProtocol="tcp" portNumber="389" portName="ldap" discover="0" />
```

to

```
<portInfo portProtocol="tcp" portNumber="389" portName="ldap" discover="1" />
```

4 Supported Versions

DFM discovers both Microsoft Exchange Server 2003 and Microsoft Exchange Server 2007 with the LDAP protocol.

5 Network and Protocols

LDAP. For an explanation, see "Prerequisites – Set Permissions" on page 129.

6 Discovery Workflow

Activate the **Microsoft Exchange Topology by LDAP** job (**Discovery Modules > Enterprise Applications > Microsoft Exchange**). This job discovers both 2003 and 2007 versions of Exchange.

7 The Microsoft Exchange Server Package

All components responsible for Exchange in DFM are bundled in the **Microsoft_Exchange_Server** package. For details on the package, click the **Readme** link in the Package Manager.

8 Additional CITs

The following CITs have been added to the Microsoft Exchange Server Package:

- Routing Group Connector
- SMTP Connector
- Exchange Routing Connector
- Send Connector
- Receive Connector
- Exchange Storage Group

- Exchange Mailbox Database
- Routing group

9 Deprecated CITs

The following CITs are deprecated; they remain in the package but are no longer reported:

- Directory Service Access DC
- Exchange Message queue
- Exchange link
- Exchange Routing Group

10 Modified CITs

The following CITs are modified:

- Exchange System is now **Exchange Organization**
- Microsoft Exchange Server includes a new attribute: **is_master**.

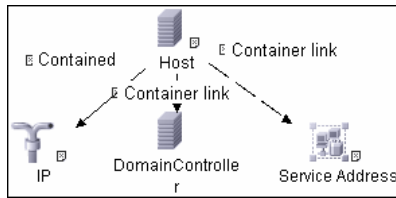
11 Discovered CITs

- Active Directory Forest
- Active Directory Site
- Active Directory System
- Administrative Group
- Contained
- Container link
- Exchange Folder
- Exchange Folder Tree
- Exchange Organization
- Exchange Routing Connector
- Exchange role
- Host

- IP
- Member
- Microsoft Exchange Server
- Routing Group Connector
- Routing group
- SMTP Connector

12 Trigger TQL

The Trigger TQL (**trigger_domainctl_ldap**) is part of the Active Directory package.



13 Trigger CI

DomainController

14 CI Attributes

CI	Attribute Value
IP	NOT IP Probe Name Is null
DomainController	NOT Reference to the credentials entry dictionary Is null AND NOT Application IP Is null
Service Address	Name Equal ignore case ldap

15 Adapters

- ▶ **ms_exchange_topology_by_ldap**. This adapter discovers Microsoft Exchange Server, versions 2003 and 2007.

16 Topology Maps

- ▶ For the Microsoft Exchange Server 2007 topology view, see "Topology Maps" on page 124.
- ▶ For the Microsoft Exchange Server 2003 topology view, see "Topology Map" on page 120.

17 Troubleshooting and Limitations

This section describes troubleshooting and limitations for Microsoft Exchange discovery.

- ▶ Currently Exchange Folders are not reported through the **Microsoft Exchange Topology by LDAP** job.

13

Microsoft MQ (Message Queue)

Note: This functionality is available as part of Content Pack 6.00 or later.

This chapter includes:

Tasks

- ▶ Discover Microsoft MQ on page 135

Reference

- ▶ Topology Discovery Methodology on page 139
- ▶ Added Entities on page 149

Tasks

Discover Microsoft MQ

The Microsoft Message Queue (MS MQ) discovery process enables you to discover MS MQ topology running with Active Directory as well as the end configuration of all MS MQ servers.

This task includes the following steps:

- ▶ "Supported Versions" on page 136
- ▶ "Discovery Workflow" on page 136
- ▶ "Scripts" on page 137

- "Trigger TQLs" on page 137
- "Input TQLs" on page 138
- "Performance" on page 139

1 Supported Versions

- Integrated Active Directory
- Stand alone
- MS MQ version 3.0 or later

2 Discovery Workflow

Activate the jobs in the following order:

- **Host Connection by Shell**
- **Host Resources and Applications by Shell**

At this stage, the CMDB contains information regarding the MS MQ Manager and machine with the domain controller on condition that the server is a member of the domain.

- **Active Directory Connection by LDAP**

This job detects which LDAP credentials are needed for discovery for the **Microsoft Message Queue Topology by LDAP** job.

- **Microsoft Message Queue Topology by NTCMD**

Discovers the server side topology (queues, triggers, rules).

- **Microsoft Message Queue Topology by LDAP**

Discovers the Active Directory topology (forest, site, site-link).

For details on how DFM discovers MQ topology, see "Topology Discovery Methodology" on page 139.

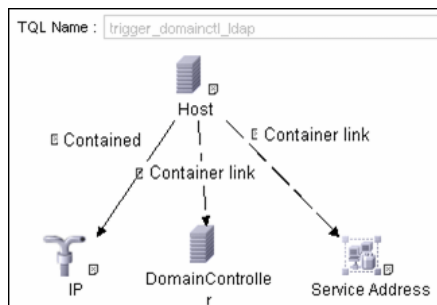
3 Scripts

To view the scripts: **Adapter Management > Discovery Packages > Microsoft_MQ > Scripts.**

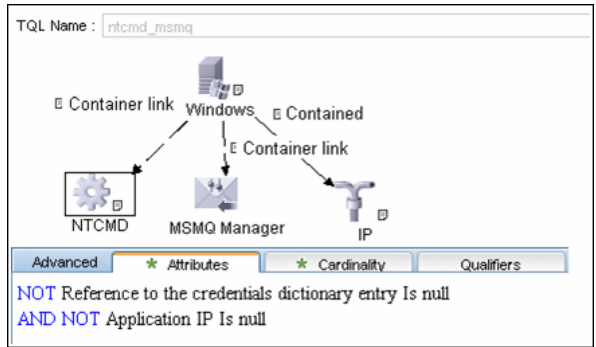
Script	Description
ntcmd_msmq.py	Main script for the Microsoft Message Queue Topology by NTCMD job
ldap_msmq.py	Main script for the Microsoft Message Queue Topology by LDAP job
plugin_microsoft_mq.py	Shallow plug-in for MS MQ Manager discovery (Adapter Management > Discovery Packages > Host_Resources_Basic > Scripts)
host_resolve_utils.py	DNS resolving utilities (Adapter Management > Discovery Packages > Host_Resources_Basic > Scripts)

4 Trigger TQLs

Microsoft Message Queue Topology by LDAP:

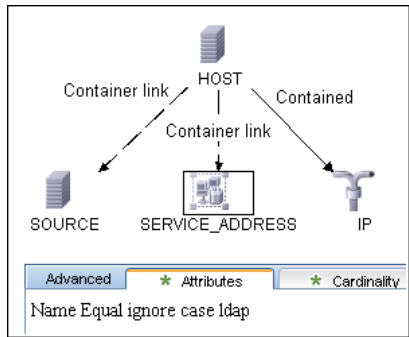


Microsoft Message Queue Topology by NTCMD:

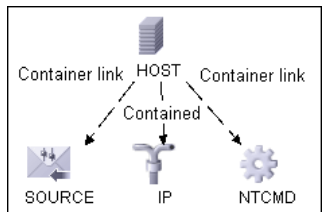


5 Input TQLs

Microsoft Message Queue Topology by LDAP:



Microsoft Message Queue Topology by NTCMD:



6 Performance

As information is retrieved from configuration files in three short registry branches only, and each file is less than 2 KB, system performance should not be affected.

Reference

Topology Discovery Methodology

This section describes how DFM discovers the MS MQ topology.

This section includes the following topics:

- "Host Resources and Applications by Shell Job" on page 139
- "Microsoft Message Queue Topology by NTCMD Job" on page 142
- "Microsoft Message Queue Topology by LDAP Job" on page 148

Host Resources and Applications by Shell Job

This job uses the `plugin_microsoft_mq.py` script.

Information is parsed from the following branches:

Registry Branch (1)

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache\
```

► **Command Output**

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache
EnterpriseId REG_BINARY C209A2FE9203F64CB543441CC92A40DC
SiteId REG_BINARY FB7BA54DFF5F40429ECA64752D0130A0
MQS_DepClients REG_DWORD 0x0
MQS REG_DWORD 0x1
MQS_DsServer REG_DWORD 0x0
MQS_Routing REG_DWORD 0x1
QMId REG_BINARY 1D19B008D7BF654B84050FC7353F993C
MachineQuota REG_DWORD 0x100000
MachineJournalQuota REG_DWORD 0xffffffff
LongLiveTime REG_DWORD 0x54600
```

► **Regular Expression Patterns**

Message routing enabled:

```
"\s*MQS_Routing\s+REG_DWORD\s+0x[0]*(\d)\s**"
```

Message storage limit:

```
"\s*MachineQuota\s+REG_DWORD\s+(\w+)\s**"
```

Message journal limit:

```
"\s*MachineJournalQuota\s+REG_DWORD\s+(\w+)\s**"
```

Registry Branch (2)

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\setup\
```

► Command Output

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\setup
MachineDomain REG_SZ UCMDB-EX
MachineDomainFQDN REG_SZ ucmdb-ex.dot
OSType REG_DWORD 0x500
CreateMsmqObj REG_DWORD 0x0
UserSid REG_BINARY
10500000000000515000000576A62162631895C45612C98F4010000
MachineDN REG_SZ CN=MSMQ-VM01,CN=Computers,DC=ucmdb-
ex,DC=dot
JoinStatus REG_DWORD 0x2
MSMQAddedToICFExceptionList REG_DWORD 0x1
MQDSSvcInstalled REG_DWORD 0x1
InetpubWebDir REG_DWORD 0x1
```

► Regular Expression Patterns

Machine domain name:

```
"s*MachineDomainFQDN\s+REG_SZ\s+([\w\-.]+)\s**"
```

Registry Branch (3)

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Setup\
```

► Command Output

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Setup
msmq_Core REG_DWORD 0x1
msmq_LocalStorage REG_DWORD 0x1
msmq_ADIntegrated REG_DWORD 0x1
InstalledComponents REG_DWORD 0xf8000000
msmq_MQDSService REG_DWORD 0x1
msmq_TriggersService REG_DWORD 0x1
msmq_HTTPSupport REG_DWORD 0x1
msmq_RoutingSupport REG_DWORD 0x1
```

► Regular Expression Patterns

MsMQ is a domain member:

```
"s*msmq_ADIntegrated\s+REG_DWORD\s+0x[0]*(\d)\s**"
```

Triggers enabled:

```
"\s*msmq_TriggersService\s+REG_DWORD\s+0x[0]*(\d)\s**"
```

Microsoft Message Queue Topology by NTCMD Job

This job discovers the settings and relationships of triggers, rules, and queues.

MS MQ Queue Discovery

► Registry Branch

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters /v  
StoreReliablePath
```

► Command Output

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters  
StoreReliablePath REG_SZ C:\WINDOWS\system32\msmq\storage
```

► Regular Expression Patterns

Base parent folder for message storage

```
"\s*StoreReliablePath\s+REG_SZ\s+(.)"
```

► Command

```
dir /B /A:-D <ms mq queue settings folder>
```

► Command Output

```
dir /B /A:-D C:\WINDOWS\system32\msmq\storage\lqs  
00000002.990736e8  
00000003.6ab7c4b8  
00000004.4c1eb11b  
00000006.e2f46f06  
00000010.d1c14377  
00000012.e6d243aa  
9b0b035bf61b429d845bbd61740403b7.0d0d6ec1
```

► Result

The file names of MS MQ queue configurations are retrieved. DFM then iterates against this list of files, reads them, and parses the queue settings.

► Command

```
type <full_path_to_the_file>
```

► Command Output

```
type C:\WINDOWS\system32\msmq\storage\lqs\00000002.990736e8

[Properties]
Label=private\admin_queue$
Type=00000000-0000-0000-0000-000000000000
QueueName=\private\admin_queue$
Journal=00
Quota=4294967295
Security=010007805c000000680000000000000014000000200480003000000000
018003f000e00010200000000000520000000200200000000140024000200010100
0000
000001000000000000140004000000010100000000000507000000010100000000
0005120000000101000000000000512000000
JournalQuota=4294967295
CreateTime=1259681363
BasePriority=32767
ModifyTime=1259681363
Authenticate=00
PrivLevel=1
Transaction=00
SystemQueue=01
Signature=DoronJ
```

► Parse Rules

Queue name:

```
".*QueueName\s*=\s*(.+?)\n.*"
```

Is transactional:

```
".*Transaction\s*=\s*(\d+).*"
```

Queue type (public/private):

```
"^[\\]*(private).*$" against Queue name
```

Message limit:

```
".*\\s+Quota\\s*=\\s*(\\d+).*"
```

Is journal enabled:

```
".*Journal\\s*=\\s*(\\d+).*"
```

Journal limit:

```
".*JournalQuota\\s*=\\s*(\\d+).*"
```

MS MQ Triggers Discovery

► Registry Branch

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\
```


► Command Output

```

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\31
b8e2c4-f412-431e-9b2c-517f7e5031d7
  Name REG_SZ Test Trigger
  Queue REG_SZ msmq-vm2\Test Queue
  Enabled REG_DWORD 0x1
  Serialized REG_DWORD 0x0
  MsgProcessingType REG_DWORD 0x1

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\31
b8e2c4-f412-431e-9b2c-517f7e5031d7\AttachedRules
  Rule0 REG_SZ 9c172d69-c832-453e-826b-4415b7d0dfef

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\72
8b0d45-531d-4887-9762-3191b0069bb1
  Name REG_SZ remote Trigger
  Queue REG_SZ msmq-vm01\Test Queue
  Enabled REG_DWORD 0x1
  Serialized REG_DWORD 0x0
  MsgProcessingType REG_DWORD 0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\72
8b0d45-531d-4887-9762-3191b0069bb1\AttachedRules
  Rule0 REG_SZ 9c172d69-c832-453e-826b-4415b7d0dfef

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\b9
00d598-e3c2-4958-bf21-c8c99ed264e2
  Name REG_SZ qqqqqqq
  Queue REG_SZ msmq-vm2\private$\Private Test Queue
  Enabled REG_DWORD 0x1
  Serialized REG_DWORD 0x0
  MsgProcessingType REG_DWORD 0x1

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\b9
00d598-e3c2-4958-bf21-c8c99ed264e2\AttachedRules
  Rule0 REG_SZ 9c172d69-c832-453e-826b-4415b7d0dfef

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\dc
4302f0-d28c-40e4-a19a-492dcee231fe
  Name REG_SZ Test2
  Queue REG_SZ msmq-vm2\private$\Test Transactional
  Enabled REG_DWORD 0x1
  Serialized REG_DWORD 0x1
  MsgProcessingType REG_DWORD 0x2

```

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\dc
4302f0-d28c-40e4-a19a-492dcee231fe\AttachedRules
  Rule0 REG_SZ 9c172d69-c832-453e-826b-4415b7d0dfef
  Rule1 REG_SZ 2874c4c1-57f1-4672-bbdd-0c16f17788cf
```

MS MQ Rule Discovery

► Regular Expression Patterns

The output buffer is split by the following regular expression:

```
"(HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\[
0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12})\s*\n"
```

After each string buffer is split, the following patterns are applied:

Trigger name:

```
".*Name\s+REG_SZ\s+(.*?)\n.*"
```

Trigger GUID:

```
" HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\
Data\Triggers\[([0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-
F]{12})\s*\n"
```

Assigned queue:

```
".*Queue\s+REG_SZ\s+(.*?)\n.*"
```

Trigger is serialized:

```
".*Serialized\s+REG_DWORD\s+0x(\d+).*"
```

Trigger is enabled:

```
".*Enabled\s+REG_DWORD\s+(0x\d+).*"
```

Trigger message processing type:

```
".*MsgProcessingType\s+REG_DWORD\s+(0xd+).**"
```

Trigger assigned rule GUID:

```
".*Rule\d+\s+REG_SZ\s+([0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}).**"
```

► Registry Branch

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\
```

► Command Output

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\2874
c4c1-57f1-4672-bbdd-0c16f17788cf
  Name REG_SZ Test Rule2
  Description REG_SZ bla bla
  ImplementationProgID REG_SZ MSQMTriggerObjects.MSMQRuleHandler
  Condition REG_SZ $MSG_PRIORITY_EQUALS=1
  $MSG_LABEL_DOES_NOT_CONTAIN=bla
  Action REG_SZ EXE C:\WINDOWS\system32\calc.exe
  ShowWindow REG_DWORD 0x1

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\9c17
2d69-c832-453e-826b-4415b7d0dfef
  Name REG_SZ Test Rule
  Description REG_SZ
  ImplementationProgID REG_SZ MSQMTriggerObjects.MSMQRuleHandler
  Condition REG_SZ $MSG_LABEL_CONTAINS=Test
  Action REG_SZ EXE C:\WINDOWS\notepad.exe
  ShowWindow REG_DWORD 0x1
```

► Regular Expression Patterns

The output buffer is split by the following constant:

```
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\"
```

After each string buffer is split, the following patterns are applied:

Rule name:

```
".*Name\s+REG_SZ\s+(.*)\n.*"
```

Rule condition:

```
".*Condition\s+REG_SZ\s+(.*)\n.*"
```

Rule action:

```
".*Action\s+REG_SZ\s+(.*)\n.*"
```

Rule GUID:

```
"\s*{[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}).*"
```

Microsoft Message Queue Topology by LDAP Job

This job reports the Active Directory-related part of MS MQ deployment: AD Forest, AD Site, MS MQ Manager, and MS MQ Routing Link.

Schema parameters:

```
CN=Configuration,DC=<domain_name>,DC=<domain_suffix>
```

Site discovery (derived from AD discovery):

```
CN=Sites,CN=Configuration,<domain_name>,DC=<domain_suffix>
```

Servers Discovery with MS MQ Manager

► Branch

```
CN=Servers,CN=<site_name>,CN=Sites,CN=Configuration,DC=<domain_name>,DC=<domain_suffix>
```

► Values

Server name property:

```
'name'
```

Server full DN:

```
'distinguishedName'
```

If an underlying branch exists (for objectClass=mSMQSettings), the server is considered to include an MS MQ Manager.

Added Entities

The following entities have been added to UCMDB:

Entity Type	Changed Entity
CI Type	Messagingsoftware
CI Type	Mqresource
CI Type	Msmqmanager
CI Type	Msmqueue
CI Type	Msmroutinglink
CI Type	Msmrule
CI Type	Msmtrigger
Attribute type definition	MessageProcessingTypeEnum
Type definition	MsMqManagerInstallationType
Type definition	MsMqQueueTypeEnum
Link	clientserver.msmqmanager.msmqmanager

Entity Type	Changed Entity
Link	contained.msmqroutinglink.mqueueemanager
Link	contained.msmqroutinglink.msmqmanager
Link	container_f.activedirectoryforest.msmqroutinglink
Link	container_f.msmqueue.msmqtrigger
Link	member.msmqroutinglink.activedirectoriesite
Link	use.msmqtrigger.msmqrule
Job	Microsoft Message Queue Topology by LDAP
Job	Microsoft Message Queue Topology by NTCMD

14

SAP

This chapter includes:

Concepts

- ▶ SAP Discovery Overview on page 152

Tasks

- ▶ Discover SAP ABAP on page 152
- ▶ Discover SAP Solution Manager on page 157
- ▶ Discover SAP Java on page 160
- ▶ Troubleshooting and Limitations on page 162

Concepts

SAP Discovery Overview

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" in *HP Universal CMDB Data Flow Management Guide*.

The Trigger CI for the following jobs is **SAP ABAP Application Server CI**:

- SAP Solution Manager Topology by SAP JCO
- SAP Solution Manager by SAP JCO
- SAP Applications by SAP JCO
- SAP ABAP Topology by SAP JCO

Tasks

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:

- "Supported Versions" on page 153
- "Prerequisites – Install Java Connectors" on page 153

- "Network and Protocols" on page 154
- "Discovery Workflow" on page 154
- "Configure Adapter Parameters" on page 156
- "Discovered CITs" on page 157

1 Supported Versions

SAP BASIS and SAP AS (Architecture layer). Versions 3.x to 6.x.

SAP JCo. Version 2.x (recommended). Note that DFM can discover SAP as long as the default SAP JCo provided with DFM is the correct version. If you are running an older version of SAP JCo, DFM may not be able to connect to SAP version 6.x.

SAP J2EE client. The version should match the relevant SAP system version.

2 Prerequisites – Install Java Connectors

- a 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

- b Extract **sapjco-ntintel-2.0.8.zip** to a temporary directory (for example: C:\temp) on the HP Universal CMDB machine.
- c Copy **sapjco.jar** from the temporary directory to the C:\hp\UCMDB\DataFlowProbe\content\lib\ directory on the machine where the Data Flow Probe is installed.
- d Copy **sapjcorfc.dll** from the temporary directory to the %winnt%\system32 directory on the machine where the Data Flow Probe is installed. Also copy the file to the C:\hp\UCMDB\DataFlowProbe\content folder.
- e Copy **librfc32.dll** from the temporary directory to the %winnt%\system32 directory. Also copy the file to the C:\hp\UCMDB\DataFlowProbe\content folder.
- f Verify that the **MSVCR71.dll** and **MSVCP71.dll** files are located in the %winnt%\system32 directory.

- g** If the Data Flow Probe is installed on a 64-bit machine on a Windows platform, place the standard **librfc32.dll** and **sapjcorfc.dll** drivers under the Windows installation folder (for example, **C:\windows\SysWOW**).

Place the **msvcp71.dll** and **msvcr71.dll** drivers under the Probe installation directory: **C:\hp\UCMDB\DataFlowProbe\content\dll**.

These drivers usually exist on a 32-bit machine and can be copied to the 64-bit machine.

3 Network and Protocols

The following protocols enable connection to a machine to verify whether a SAP system is installed on it:

- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SSH.** For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SAP.** For credentials information, see "SAP Protocol" in *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

- a** In the Discovery Control Panel window, activate the modules in the following order:
 - ▶ **Network – Basic** (Range IPs by ICMP, Host Connection By Shell).
 - ▶ **Host Resources and Applications** (Host Resources and Applications by Shell). This job discovers SAP running software 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.

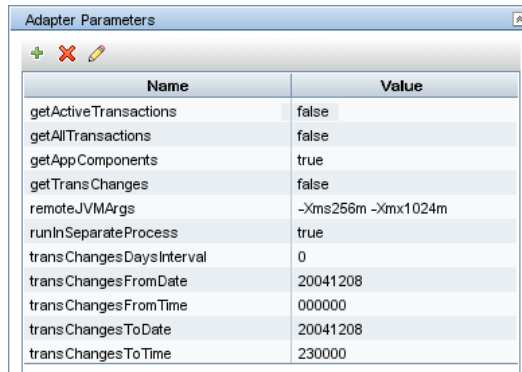
- ▶ **Web Servers – Basic** (WebServer Detection using TCP Ports). If the SAP system has an ITS configuration, to discover the ITS entities of the SAP system, run this job as a prerequisite to the SAP discovery that discovers ITS entities.
- ▶ **Application – SAP**
 - ▶ **SAP System By Shell**. This 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 adapter parameters. For details, see "Configure Adapter Parameters" on page 156.
 - ▶ **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. For details, see "Discover SAP Solution Manager" on page 157.
- b** For details on the CIs that are discovered, see the Statistics table in the Details tab, or click the **View CIs in Map** button. For details, see "Discovery Job Details Pane" in *HP Universal CMDB Data Flow Management Guide*.

- c Verify that DFM 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" and "Discovered CIs Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.

5 Configure Adapter Parameters

To omit unnecessary CIs, you can configure the adapter parameters, as follows:

- a Access the SAP adapter: **Adapter Management > SAP_discovery package > Adapters > SAP_Dis_Applications**.
- b Select the **Adapter Definition** tab and locate the **Adapter Parameters** pane.

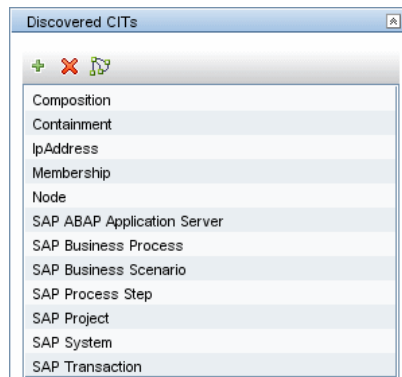


Name	Value
getActiveTransactions	false
getAllTransactions	false
getAppComponents	true
getTransChanges	false
remoteJVMArgs	-Xms256m -Xmx1024m
runInSeparateProcess	true
transChangesDaysInterval	0
transChangesFromDate	20041208
transChangesFromTime	000000
transChangesToDate	20041208
transChangesToTime	230000

- c Set one of the following parameters, and click **OK** to save the changes:
 - To discover all SAP transactions: Set **getAllTransactions** to **true**.
 - 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 MM/DD/YYYY or YYYYMMDD.
 - Set the from time (**transChangesFromTime**) and the to time (**transChangesToTime**). The time format is HH:MM:SS or HHMMSS.

6 Discovered CITs



Discover SAP Solution Manager

Note: This functionality is available as part of Content Pack 2.00 or later.

Often, an environment includes more than one SAP system, each one using a different set of credentials (for instance, user name, password, system number, or client number).

It is customary to register all SAP systems in the SAP Solution Manager, to centralize the management of the SAP systems. DFM enables discovery of all the SAP systems by discovering this connection to the SAP Solution Manager. In this way, you create a single set of credentials; there is no need to create a set of credentials for each SAP system. DFM discovers all systems (and their topology) with this one set.

DFM discovers the SAP business layer (with the **SAP Solution Manager by SAP JCO** job) and the complete topology of registered SAP systems (with the **SAP Solution Manager Topology by SAP JCO** job).

This task includes the following steps:

- "Prerequisites" on page 158
- "Supported Versions" on page 159
- "Network and Protocols" on page 159
- "Discovery Workflow" on page 159
- "Discovered CITs" on page 160
- "Topology Map" on page 160

1 Prerequisites

To run SAP Solution Manager, ask the SAP Solution Manager admin to give you permissions on the following objects for the given profile:

- For the **S_RFC** object, obtain privileges: RFC1, SALX, SBDC, SDIF, SDIFRUNTIME, SDTX, SLST, SRFC, STUB, STUD, SUTL, SXMB, SXMI, SYST, SYSU, SEU_COMPONENT.
- For the **S_XMI_PROD** object, obtain:

```
EXTCOMPANY=MERCURY;EXTPRODUCT=DARM;INTERFACE=XAL
```

- For the **S_TABU_DIS** object, obtain:

```
DICBERCLS=SS; DICBERCLS=SC; DICBERCLS=&NC& ACTVT=03
```

2 Supported Versions

SAP Solution Manager versions 6.x, 7.x.

3 Network and Protocols

SAP. For credentials information, see "SAP Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

Method 1:

- Run the **SAP TCP Ports** job to discover SAP ports.
- Run the **SAP ABAP Connection by JCO** job.
- Run the **SAP Solution Manager Topology by SAP JCO** job.
- Run the **SAP Solution Manager by SAP JCO** job.

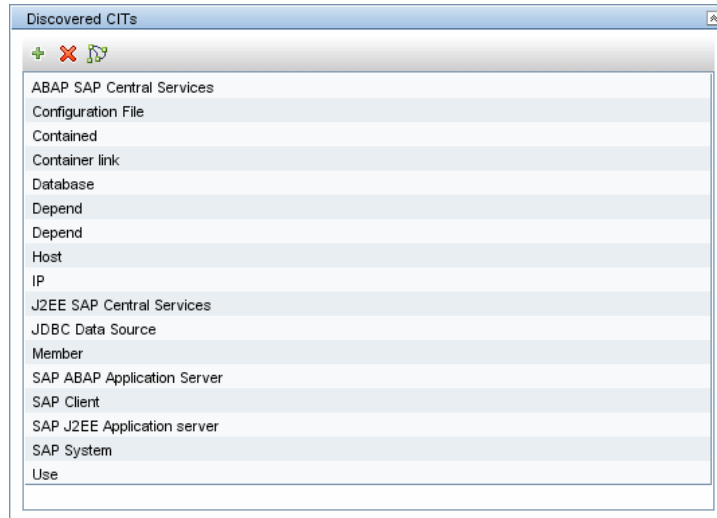
Method 2:

- Run the **Host Resources by ...** jobs to discover SAP (ABAP or J2EE) Application Server and/or SAP (ABAP or J2EE) Central Services.
- Run the **SAP System by Shell** job to create a SAP system CI (but without defining whether it is the SAP Solution Manager).
- Run the **SAP ABAP Connection by JCO** job.
- Run the **SAP Solution Manager Topology by SAP JCO** job.
- Run the **SAP Solution Manager by SAP JCO** job.

During the run of the SAP ABAP Connection by JCO job, the SAP Systems that are defined as the SAP Solution Manager will be triggered on these two jobs: **SAP Solution Manager Topology by SAP JCO** and **SAP Solution Manager by SAP JCO** job.

5 Discovered CITs

The following CITs are discovered by the SAP Solution Manager Topology by SAP JCO job:



6 Topology Map

To view the SAP Solution Manager Topology by SAP JCO map: Discovery Control Panel > select **Enterprise Applications > SAP > SAP Solution Manager Topology by SAP JCO > Details pane**. Click the **View CIs in Map** button.

Discover SAP Java

The SAP for Java discovery process enables you to discover SAP JAVA architecture and J2EE applications on the SAP JAVA server.

This task includes the following steps:

- "Prerequisites" on page 161
- "Network and Protocols" on page 161
- "Discovery Workflow" on page 162
- "Discovered CITs" on page 162

1 Prerequisites

- a** Add the following *.jar files to the `C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\j2ee\sap` directory on the Data Flow 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\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\j2ee\sap` directory on the Data Flow 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 Data Flow 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.

2 Network and Protocols

The following protocol enables connection to a machine and verification whether a SAP system is installed on it:

- ▶ **SAP JMX.** For credentials information, see "SAP JMX Protocol" in *HP Universal CMDB Data Flow Management Guide*.

3 Discovery Workflow

In the Discovery Control Panel window, activate the modules in the following order:

- ▶ **Network – Basic** (Range IPs by ICMP, Host Connection By Shell).
- ▶ **Host Resources and Applications** (Host Resources and Applications by Shell). This job discovers SAP running software 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.

4 Discovered CITs

Troubleshooting and Limitations

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.

15

Siebel

This chapter includes:

Concepts

- ▶ Overview on page 163

Tasks

- ▶ Discover Siebel Topology on page 164

Troubleshooting and Limitations on page 170

Concepts

Overview

Using the Siebel adapters, you can run an automatic Siebel discovery to create the Siebel world, together with its components, inside HP Universal CMDB. During discovery:

- ▶ All Siebel-related IT entities that reside in the organization are discovered and configuration items (CIs) are written to the CMDB.
- ▶ 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.

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.

Tasks

Discover Siebel Topology

This task describes how to discover Siebel topology.

This task includes the following steps:

- "Prerequisites – Copy the driver Tool to the Data Flow Probe" on page 164
- "Network and Protocols" on page 165
- "Discovery Workflow" on page 166
- "Discovered CITs" on page 167
- "Topology Map – Siebel Topology View" on page 169
- "Troubleshooting and Limitations" on page 170

1 Prerequisites – Copy the driver Tool to the Data Flow Probe

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 Data Flow Probe:

- a** Copy the driver Command Line Interface (CLI) tool from the Siebel server to any folder on the Data Flow Probe machine.
- 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 Data Flow Probe machine 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.

2 Network and Protocols

Set up the following protocols for the Windows platform:

- **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- **Siebel Gateway.** For credentials information, see "Siebel Gateway Protocol" in *HP Universal CMDB Data Flow Management Guide*.

Set up the following protocols for the UNIX platform:

- **SSH.** For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- **Telnet.** For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- **Siebel Gateway.** For credentials information, see "Siebel Gateway Protocol" in *HP Universal CMDB Data Flow Management Guide*.

3 Discovery Workflow

- a** For Siebel discovery to run, you must copy the driver tool to the Data Flow Probe server. For details, see "Prerequisites – Copy the driver Tool to the Data Flow Probe" on page 164.
- b** To trigger the discovery of Siebel networking features, add a Network CI to the CMDB. For details, see "New CI/New Related CI Dialog Box" in *Modeling Guide*.
- c** In the Discovery Control Panel 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)
- d** 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)
- e** To discover Siebel, activate all the jobs in the **Application – Siebel** module.

Note: The following enrichment adapters automatically run in the background during discovery:

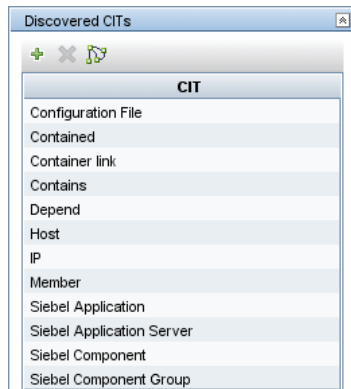
Siebel_Route_WebApp_To_Component. Builds the route between Siebel Web Application CIs and Siebel Component CIs.

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.

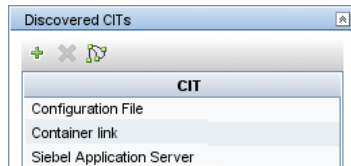
- f** For details on the CIs that are discovered, see the Statistics table in the Details tab.

4 Discovered CITs

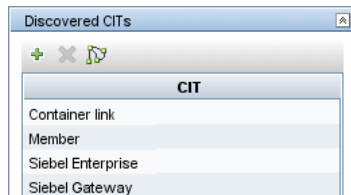
SIEBEL_DIS_APP_SERVERS:



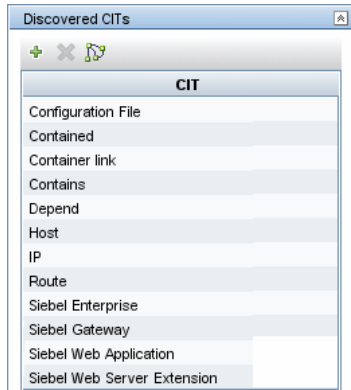
SIEBEL_DIS_APP_SERVER_CONFIG:



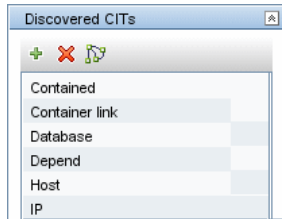
SIEBEL_DIS_GATEWAY_CONNECTION_(GTWY)



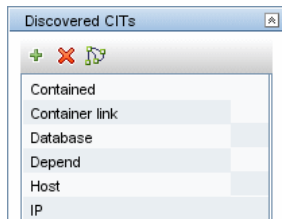
SIEBEL_DIS_WEBAPPS_UNIX:



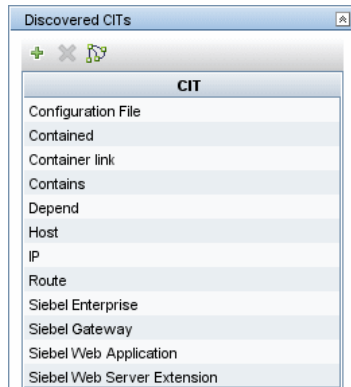
SIEBEL_DIS_DB_UNIX



SIEBEL_DIS_DB_NT



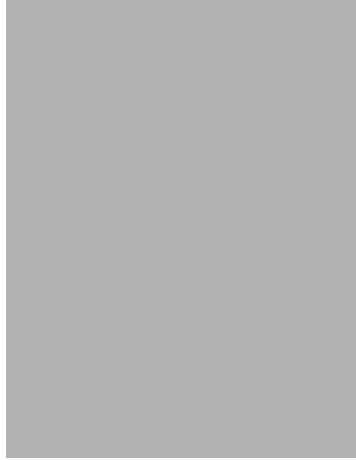
SIEBEL_DIS_WEBAPPS_NT"



5 Topology Map – Siebel Topology View



6 Topology Map – Siebel Web Topology View



Troubleshooting and Limitations

This section describes troubleshooting and limitations for Siebel discovery.

- ▶ 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.

16

UDDI Registry

This chapter includes:

Concepts

- Overview on page 171

Tasks

- Discover UDDI Processes on page 172

Concepts

Overview

The UDDI discovery process enables you to discover Web services from a UDDI registry.

DFM 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 Resource** CI in the CMDB and the registry is created as a **UDDI Registry** CI.

Tasks

Discover UDDI Processes

This task includes the following steps:

- "Supported Versions" on page 172
- "Network and Protocols" on page 172
- "Discovery Workflow" on page 172
- "Discovery Workflow – Optional" on page 173
- "Topology Map" on page 173
- "Troubleshooting and Limitations" on page 173

1 Supported Versions

DFM supports UDDI versions 2 and 3.

2 Network and Protocols

Set up the **UDDI protocol**. For credentials information, see "UDDI Registry Protocol" in *HP Universal CMDB Data Flow Management Guide*.

3 Discovery Workflow

- a** In the Discovery Control Panel window, locate the **Application – UDDI Registry** module. Activate the **WebServices by URL** job.
- b** Activate the following jobs:
 - WebServices by URL
 - Webservice Connections by UDDI Registry
 - Webservices by UDDI Registry

For details on the CIs that are discovered, see the Statistics table in the Details tab.

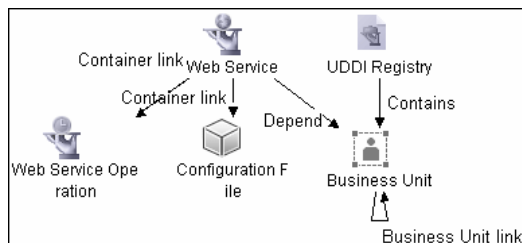
4 Discovery Workflow – Optional

To enter the name of the service publisher whose services must be published:

- a** Access the Resource Configuration window.
- b** In the Discovery Resources pane, locate the Webservices package and select the **UDDI_Registry** adapter.
- c** In the **Adapter Definition** tab, in the **Adapter 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.

5 Topology Map

The following depicts the topology of the **SOA_UDDI_View**:



6 Troubleshooting and Limitations

This section describes troubleshooting and limitations for UDDI Registry discovery.

- ▶ When activating the **WebServices by URL** job, the Web services being discovered should be authentication-less, that is, they can be accessed without credentials (user name/password).

17

WebSphere MQ

Note: This functionality is available as part of Content Pack 6.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 176

Tasks

- ▶ Discover WebSphere MQ on page 178

Reference

- ▶ Discovered CITs on page 182
- ▶ Relationships on page 185
- ▶ Enrichment Rule on page 188
- ▶ Views and Reports on page 188

Troubleshooting and Limitations on page 192

Concepts

Overview

The WebSphere MQ package enables mapping the various components of WebSphere MQ infrastructure in an organization. The end goal is to model its interdependence with other applications or services within the organization and enable end to end impact analysis across the messaging silo.

Message Queuing is a middle-ware technology that enables disparate software services to communicate in a way that does not require any knowledge of the target service. Reliable communication can be achieved regardless of current availability of the target system or complexity of the infrastructure connecting the two systems.

A Message may contain simple character data, numeric data, complex binary data, a request for information, a command, or a mixture of all of these. The messaging infrastructure is responsible for reliable and transparent transportation of a message from the source to the target and is not required to understand or be aware of its content.

Data Flow Workflow Mechanism

WebSphere MQ can be installed on several UNIX platforms and Microsoft Windows and is managed using a command line interface standardized across platforms. The command line interface is accessible through programs **runmqsc** or **runmqadm** that are included in a WebSphere MQ installation.

The **MQ by Shell** job uses the Shell CI associated with a server as its trigger. Since every server in the CMDB may have an associated Shell CI, the trigger TQL results contain the Shell CI only for servers on which WebSphere MQ software is installed.

The **MQ by Shell** job uses the WebSphere MQ command line interface to query for MQ objects and their details. Since the **runmqsc** command requires administrator or root privileges and the **runmqadm** command is not always available, the job attempts the **runmqadm -r** command first. If **runmqadm** fails, the job tries the **runmqsc** command.

After logging in to the MQ server using the Shell CI (created by the Host Connections by Shell job), DFM:

- a** Identifies the version of WebSphere MQ installed on the server. This is done using the **dspmqver** command. (If **dspmqver** fails, the **mqver** command is attempted.)
- b** Retrieves a list of WebSphere MQ Queue Managers using the **dspmqr** command.
- c** Retrieves details on each Queue Manager using the MQ CLI (command line interface) command:

```
DISPLAY QMGR DESCR DEADQ DEFXMITQ REPOS CCSID
```

- d** Retrieves a list of queues on each Queue Manager using the MQ CLI command:

```
DISPLAY QUEUE(*) TYPE DESCR CLUSTER CLUSNL USAGE RNAME  
RQMNAME XMITQ TARGQ DEFTYPE
```

Relationships between queues and other MQ objects such as other queues, Queue Managers, and so on, are built on the fly.

- e** Retrieves (for each TRANSMIT Queue found) the remote server name and IP and port using the sender channel associated with the transmit queue. This is done using the MQ CLI command:

```
DISPLAY CHANNEL(*) WHERE(xmitq EQ <transmitQueueName>) TYPE(SDR)  
CONNAME
```

- f** Retrieves a list of channels on each Queue Manager using the MQ CLI command:

```
DISPLAY CHANNEL(*) CHLTYPE TRPTYPE DESCR CLUSTER CLUSNL  
CONNAME XMITQ
```

Relationships between channels and other MQ objects such as other queues, channels, and so on, are built on the fly.

- g** Retrieves a list of clusters that each Queue Manager is a member of, or knows about, using the MQ CLI command:

```
DISPLAY CLUSQMGR(*) CONNAME QMTYPE
```

Relationships between clusters and other clusters are built on the fly.

- h** Retrieves the namelists that each Queue Manager is a member of, or knows about, using the MQ CLI command:

```
DISPLAY NAMELIST(*) NAMES NAMCOUNT DESCR
```

Tasks

Discover WebSphere MQ

The WebSphere MQ job discovers WebSphere MQ components and includes the following steps:

- "Supported Versions" on page 178
- "Network and Protocols" on page 179
- "Package Deployment" on page 179
- "Discovery Workflow" on page 179
- "Adapter Parameters" on page 180
- "Discovered CITs" on page 181

1 Supported Versions

IBM WebSphere MQ, versions 5.x, 6.x, and 7.x.

Target Platform. IBM WebSphere MQ

Target Platform Versions. 5.x, 6.x, 7.x

Target Platform OS. Microsoft Windows, Solaris, Linux, AIX

2 Network and Protocols

- **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- **SSH.** For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- **Telnet.** For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.

The Shell commands are (sudo is optional):

- **dspmqver** or **mqver**
- **dsmpq**
- **runmqsc** or **runmqadm -r**

3 Package Deployment

- a** Deploy the WebSphere MQ package. For details, see "Deploy a Package" in *HP UCMDB Administration Guide*.
- b** Populate the appropriate SSH, Telnet, or NTCMD protocol. For details, see "Network and Protocols" on page 179.
- c** Verify that all WebSphere MQ server IP addresses are within the scope of the Data Flow Probe. For details, see "Add/Edit IP Range Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.
- d** Configure parameters for the **MQ by Shell** job as necessary. For details, see "Details Tab" in *HP Universal CMDB Data Flow Management Guide*.

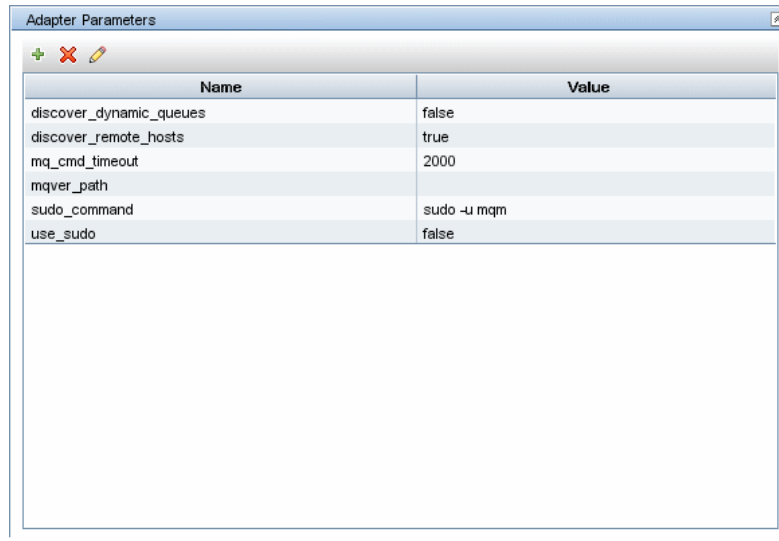
4 Discovery Workflow

Run the following jobs to collect information required to trigger WebSphere MQ discovery:

- **Range IPs by ICMP (Network Discovery – Basic).** Discovers the WebSphere MQ server IP addresses.
- **Host Connection by Shell (Network Discovery – Basic).** Discovers operating system information on the WebSphere MQ servers.

- ▶ **Host Resources and Applications by Shell (Network Discovery – Host Resources and Applications).** Discovers instances of WebSphere MQ on the servers.
- ▶ **MQ by Shell (Enterprise Applications – WebSphere MQ).** Discovers the WebSphere MQ infrastructure.

5 Adapter Parameters



The screenshot shows a window titled 'Adapter Parameters' with a table containing the following data:

Name	Value
discover_dynamic_queues	false
discover_remote_hosts	true
mq_cmd_timeout	2000
mqver_path	
sudo_command	sudo -u mqm
use_sudo	false

discover_dynamic_queues. Enables discovery of dynamic queues (Queues created and destroyed on the fly by applications).

discover_remote_hosts. Enables resolution and discovery of remote servers and MQ objects referenced by the MQ server being discovered. If set to **false**, relationships between MQ objects on different servers are not discovered.

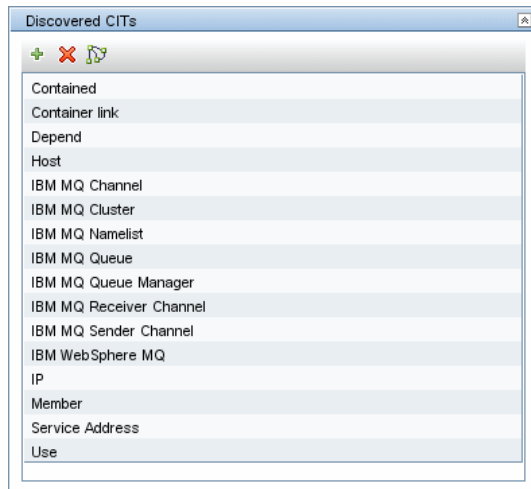
mq_cmd_timeout. Sets the command time-out for MQ CLI commands.

mqver_path. Path to **mqver** or **dspmqver** executable files. Separate multiple entries by a comma (;).

sudo_command. Must be set if the **use_sudo** parameter is set to **true**. Any entry here is prefixed to the MQ command line interface program. This parameter is typically used to set the MQ username. For example, if this parameter is set to **sudo -u mqm** the **runmqsc** command is invoked as **sudo -u mqm runmqsc**.

use_sudo. Set to **true** to enable sudo usage.

6 Discovered CITs



For details, see "Discovered CITs" on page 182.

Reference

Discovered CITs

The WebSphere MQ package contains the following CI Types:

CI Type	Key Attributes	Description
IBM WebSphere MQ (webspheremq) Parent: Message Queuing Software	<ul style="list-style-type: none"> ▶ Name: Always IBM WebSphere MQ ▶ Container: Host 	Represents an instance of WebSphere MQ software installed on a server.
IBM MQ Queue Manager (mqqueue) Parent: Message Queue Resource	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM WebSphere MQ CI 	Represents an MQ Queue Manager. A WebSphere MQ instance may have one or more Queue Managers. The Queue Manager is responsible for functions not directly related to data movement such as storage, timing, triggering, and so on. Queue managers use a proprietary IBM technology known as a bindings connection to communicate with the MQ objects it manages and with remote clients via a network.
IBM MQ Namelist (mqnamelist) Parent: Message Queue Resource	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	Represents an MQ Namelist. An MQ namelist contains a list of names and is typically used to contain a list of MQ Queue Manager Clusters. These namelists are then specified in the cluster namelist property and may be used by all Queue Managers in that cluster for look up.
IBM MQ Channel (mqchannel) Parent: Message Queue Resource	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	This abstract CI Type represents MQ Channels. MQ Channels are required by Queue Managers to communicate with other Queue Managers. Channels have uni-directional and bi-directional communication (such as a request-response system) and require a second channel to return data. A channel sends or receives data on a specific port on a TCP/IP network.

CI Type	Key Attributes	Description
IBM MQ Cluster (mqcluster) Parent: Failover Cluster	Name	Represents an MQ Queue Manager Cluster. An MQ Cluster provides a flexible approach to join multiple Queue Managers with minimal configuration. This enables multiple instances of the same service to be hosted through multiple Queue Managers, resulting in higher performance, capacity, and resiliency. Queue managers can dynamically join or leave clusters.
IBM MQ Queue (mqqueue) Parent: MQ Queue	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	A Queue is a container of messages in the MQ infrastructure and controls how messages are routed between Queue Managers in the MQ infrastructure. Queues may be set up in several configurations to control message ordering and delivery (F/LIFO, message priority, sequential delivery, guaranteed delivery, and so on) and are optimized to carry small amounts of information.
IBM MQ Alias Queue (mqlocalqueue) Parent: IBM MQ Queue	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	Represents MQ Alias Queues. An Alias Queue is an alias of another queue. It can be an alias of a local, remote, transmission, or another alias queue. The alias queue and the queue for which it is an alias are within the same Queue Manager. Messages and commands issued on the alias queue are forwarded to the queue for which it is an alias.
IBM MQ Local Queue (mqlocalqueue) Parent: IBM MQ Queue	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	Represents MQ Local Queues. A Local Queue is a basic message queue and container of messages. An application can place a message in it for delivery or request, or retrieve a message from it.

CI Type	Key Attributes	Description
IBM MQ Remote Queue (mqlocalqueue) Parent: IBM MQ Queue	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	Represents MQ Remote Queues. A Remote Queue is a remote or proxy instance of another queue. It can be a remote instance for a local, remote, transmission, or another alias queue. The remote queue and the queue for which it is a remote may be on different Queue Managers. A Remote Queue may also be a remote or proxy of a Queue Manager, and is represented as a remote Queue Manager.
IBM MQ Transmit Queue (mqlocalqueue) Parent: IBM MQ Queue	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	Represents MQ Transmission Queues. A Transmission Queue is a special purpose queue that transmits messages from one Queue Manager to another through MQ Channels. Remote queues use transmission queues to relay messages to the queue for which it is a remote.
IBM MQ Receiver Channel (mqreceiverchannel) Parent: IBM MQ Channel	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	A receiving channel receives messages from remote Queue Managers through a sending channel with the same name.
IBM MQ Sender Channel (mqsenderchannel) Parent: IBM MQ Channel	<ul style="list-style-type: none"> ▶ Name ▶ Container: IBM MQ Queue Manager 	A sending channel is associated with a specific Transmission queue within the same parent Queue Manager and has a well-defined destination.



Relationships

The WebSphere MQ package contains the following relationships:

Link	End1	End2	Cardinality	Description
Client Server	IBM MQ Send Channel	IBM MQ Receive Channel	1..*	Represents the direction of message flow between MQ Channels
Realization	IBM MQ Remote Queue	IBM MQ Queue	1..*	Indicates a strong dependency between an MQ Remote Queue and another Queue for which it is a remote. This is used in situations when the type of Queue is unknown.
Realization	IBM MQ Remote Queue	IBM MQ Local Queue	1..*	Indicates a strong dependency between an MQ Remote Queue and a Local Queue for which it is a remote.
Realization	IBM MQ Remote Queue	IBM MQ Alias Queue	1..*	Indicates a strong dependency between an MQ Remote Queue and an Alias Queue for which it is a remote.
Realization	IBM MQ Remote Queue	IBM MQ Remote Queue	1..*	Indicates a strong dependency between an MQ Remote Queue and a Remote Queue for which it is a remote.
Realization	IBM MQ Alias Queue	IBM MQ Queue	1..*	Indicates a strong dependency between an MQ Alias Queue and another Queue for which it is an alias. This is used in situations when the type of Queue is unknown.
Realization	IBM MQ Alias Queue	IBM MQ Local Queue	1..*	Indicates a strong dependency between an MQ Alias Queue and a Local Queue for which it is an alias.

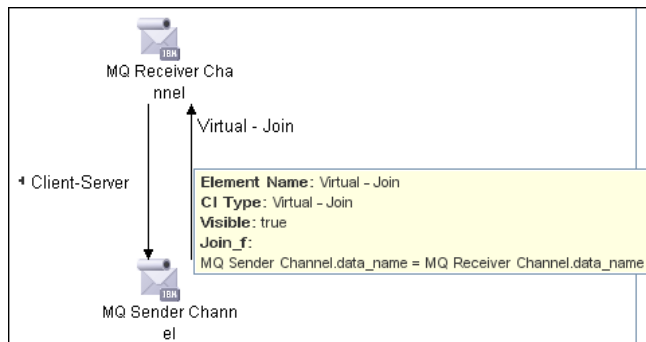
Link	End1	End2	Cardinality	Description
Realization	IBM MQ Alias Queue	IBM MQ Remote Queue	1..*	Indicates a strong dependency between an MQ Alias Queue and a Remote Queue for which it is an alias.
Realization	IBM MQ Alias Queue	IBM MQ Alias Queue	1..*	Indicates a strong dependency between an MQ Alias Queue and an Alias Queue for which it is an alias.
Realization	IBM MQ Remote Queue	IBM MQ Queue Manager	1..*	Relates a queue of type remote queue (Remote Queue Manager) and the Queue Manager it is representing. This is a special purpose Remote Queue that is a remote for Queue Manager (instead of a remote queue). For Queue Managers QM1 and QM2, it is possible to set up a Remote Queue on QM1 named RQM2 which is a remote of QM2. Any MQ command issued to RQM2 is passed on to QM2 for execution.
Member	IBM MQ Cluster	IBM MQ Queue Manager	1..*	Indicates that the MQ Queue Manager is a member of the MQ Queue Manager Cluster. If an MQ Queue Manager is a full repository for a cluster, the name of this relationship is set to Repository .

Link	End1	End2	Cardinality	Description
Member	IBM MQ Cluster	IBM MQ Channel	1..*	Indicates that the MQ Channel is a member of the MQ Queue Manager Cluster. When a queue or channel is defined in any Queue Manager, it is possible (but not necessary) to specify of which MQ cluster this queue is a member. This is useful when very specific configurations are required, for example, when a queue is a member of a cluster but the Queue Manager is not a member of that cluster. This link is used to identify these special configurations.
Member	IBM MQ Cluster	IBM MQ Queue	1..*	Indicates that the MQ Queue is a member of the MQ Queue Manager Cluster. This link is added for the same reason as in the previous row.
Member	IBM MQ Namelist	IBM MQ Channel	1..*	Indicates that the MQ Channel contains the name of the MQ Namelist in its CLUSNL parameter.
Member	IBM MQ Namelist	IBM MQ Queue	1..*	Indicates that the MQ Queue contains the name of the MQ Namelist in its CLUSNL parameter.
Use	IBM MQ Cluster	IBM MQ Channel	1..*	Indicates the MQ Channel (of types Cluster Sender Channel or Cluster Receiver Channel) used by the MQ Queue Manager Cluster for communication with another cluster. This relationship is specific to MQ Channels of type Cluster Sender Channel and Cluster Receiver Channel. These channels are dedicated to inter-cluster communication and are not used by queues or other MQ objects.

Link	End1	End2	Cardinality	Description
Use	IBM MQ Remote Queue	IBM MQ Transmit Queue	1..*	Indicates a remote queue using a transmission queue for communication.
Use	IBM MQ Transmit Queue	IBM MQ Sender Channel	1..*	Indicates a sender Transmission Queue using a Sender channel for communication.

Enrichment Rule

The WebSphere MQ package includes an enrichment rule to link sender and receiver channels. The sender and receiver channels reside on different Queue Managers and have the same name.

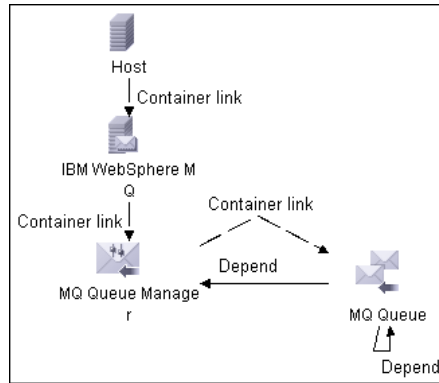


Views and Reports

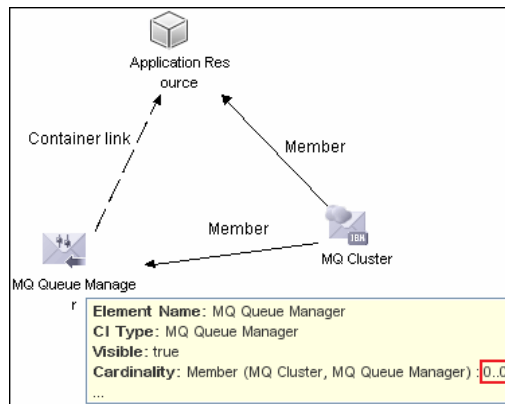
The WebSphere MQ package includes the following views that model details of the MQ infrastructure. Each view has a corresponding report with the same TQL configuration.

Note: The following out-of-the-box views are provided as examples only. You may prefer to define your own views.

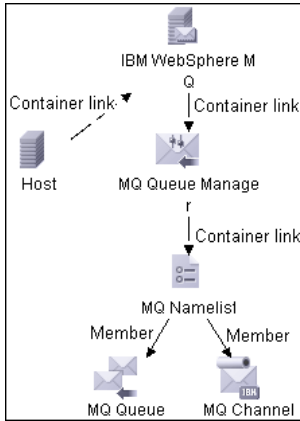
MQ Queue Dependency. This view displays queues that are dependent on other MQ objects and typically include Remote Queues, Alias Queues, and Remote Queue Managers:



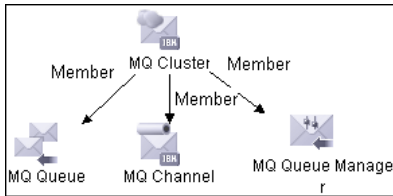
MQ Q Manager Resources on non-local Cluster. This view displays MQ objects managed by a Queue Manager and belonging to an MQ Cluster that the Queue Manager is not a member of. Any MQ objects in this view may be misconfigured and the purpose of this view is to identify such misconfigured objects.



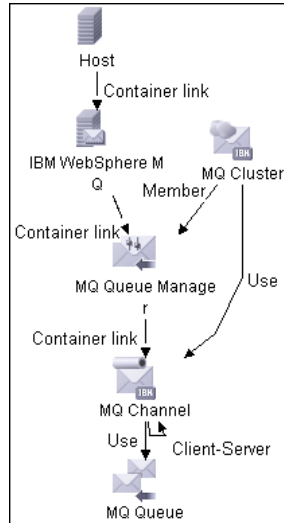
MQ Namelist Membership. This view displays namelists and their members:



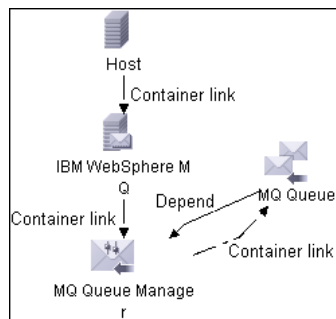
MQ Cluster Membership. This view displays clusters and their members:



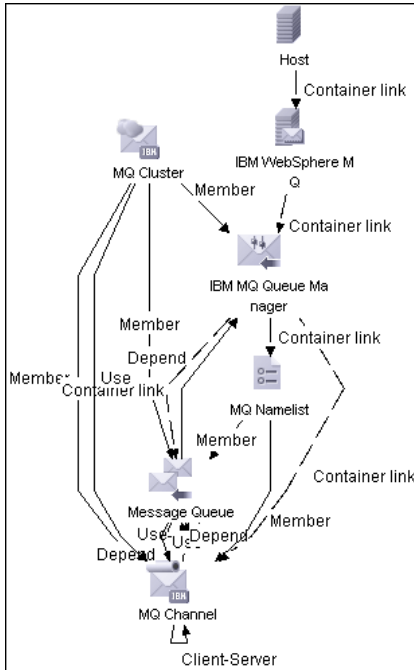
MQ Channel Communication. This view displays client-server communication between MQ Channels and queues used by the channels:



MQ Alias Queue Managers. This view displays Queues that are serving as remote Queue Managers:



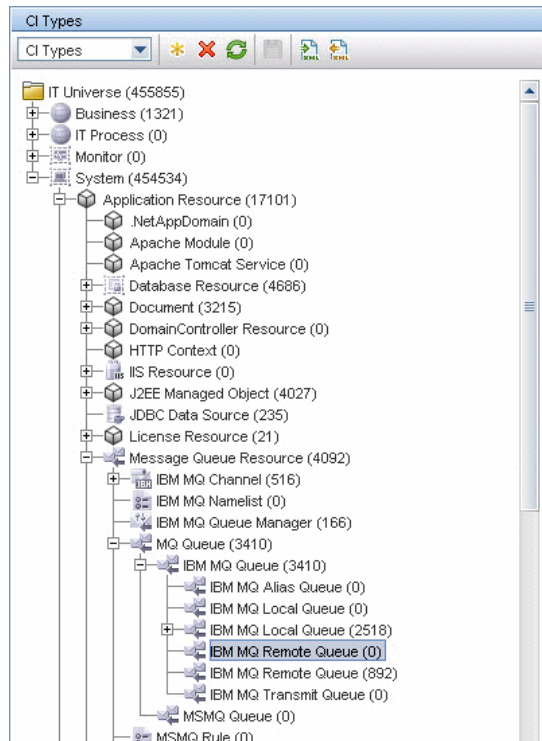
MQ Topology. This view displays all MQ objects in the MQ infrastructure including relationships and interdependencies:



Troubleshooting and Limitations

- ▶ If there are DNS resolution errors in the log files and discovery takes abnormally long to complete, try setting the **discovery_remote_hosts** parameter to **false**. For details, see "Adapter Parameters" on page 180.
- ▶ If the discovery results appear incomplete, try increasing the value of the **mq_cmd_timeout** parameter. For details, see "Adapter Parameters" on page 180.

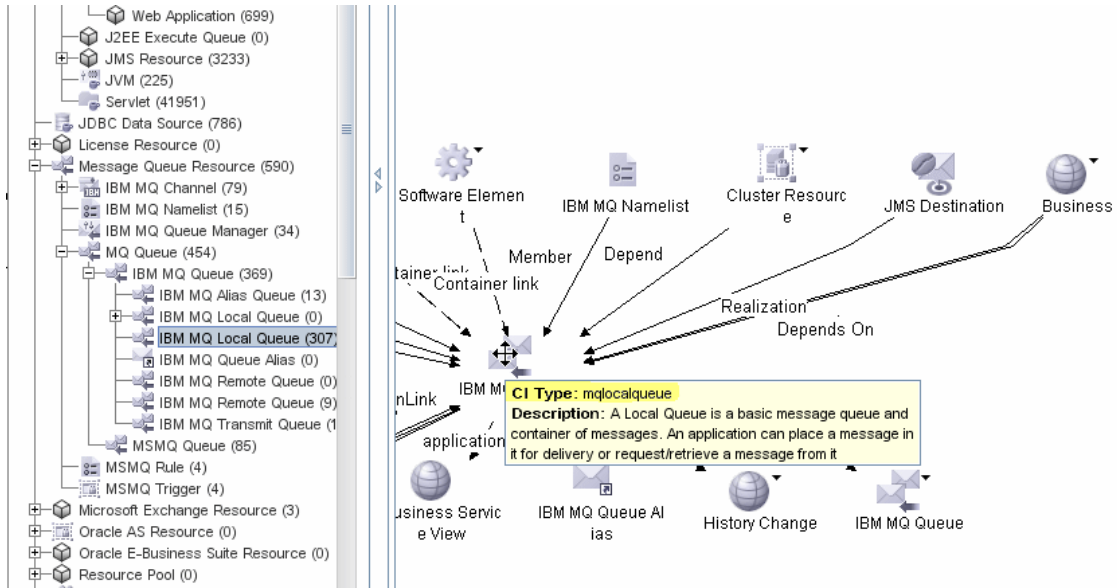
- Two instances of the **IBM MQ Local Queue** and **IBM MQ Remote Queue** CITs are displayed in the CI Type Manager:



This is because the CIT name changed for these CITs between Content Pack 5 and Content Pack 6, from **mqueuelocal** and **mqueueremote** to **mqlocalqueue** and **mqremotequeue**.

The Content Pack 6 jobs populate the correct CITs (**mqlocalqueue** and **mqremotequeue**). You should create reports, view, and so on using these CITs.

Hold the cursor over the CIT to view the CIT name:



18

JBoss

This chapter includes:

Concepts

- ▶ JBoss Discovery Overview on page 195

Tasks

- ▶ Discover JBoss by JMX on page 196
- ▶ Discover JBoss by Shell on page 198

Concepts

JBoss Discovery Overview

This section describes how to discover JBoss applications. The JBoss discovery process enables you to discover a full JBoss topology including J2EE applications, JDBC, and JMS resources.

DFM first finds JBoss servers based on the JMX protocol, then discovers the JBoss J2EE environment and components.

Tasks

Discover JBoss by JMX

This task includes the following steps:

- "Prerequisites" on page 196
- "Supported Versions" on page 196
- "Network and Protocols" on page 196
- "Discovery Workflow" on page 197
- "Adapter Parameters for JBoss by JMX" on page 197
- "Discovered CITs" on page 197

1 Prerequisites

- a Run the **Range IPs by ICMP** job.
- b Set up the drivers needed to discover JBoss. Default JBoss drivers are included by default with the Probe installation. For details on the required *.jar files, see "JBoss" in *HP Universal CMDB Data Flow Management Guide*.

The Probe installation includes JBoss drivers for versions 3.x and 4.x, but you can use your own drivers, if you prefer.

The *.jar files needed in discovery are located in the following folder:

```
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\
discoveryResources\j2ee\jboss\<version number>.x\
```

2 Supported Versions

JBoss versions 3.x, 4.x.

3 Network and Protocols

JBoss. For credentials information, see "JBoss Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

Run the following jobs:

- J2EE TCP Ports
- JBoss Connections by JMX
- JBoss by JMX

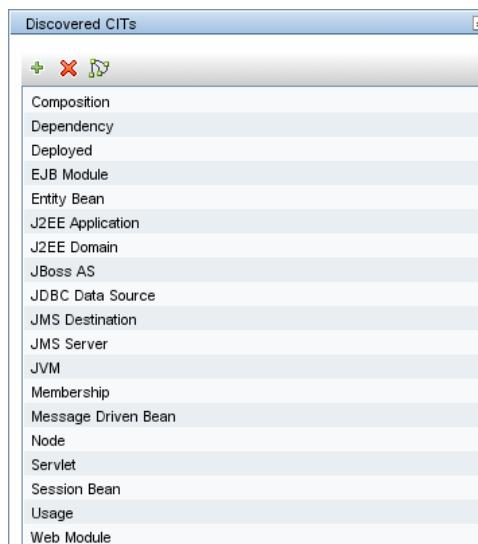
5 Adapter Parameters for JBoss by JMX

The following adapters determine whether JMS and Application Resources related components can be omitted from reports to the UCMDB server:

- **discoverAppResources**. **True** or missing (as previously): a full application deployment discovery is performed. **False**: only the application is discovered without all its resources.
- **discoverJMSResources**. **True** or missing (as previously): A full JMS discovery is performed. **False**: no JMS discovery is performed.

6 Discovered CITs

- The following CITs are discovered by the **JBoss by JMX** job:



Discover JBoss by Shell

Note: This functionality is available as part of Content Pack 2.00 or later.

You can perform deep discovery of JBoss without having to enter JMX credentials for each server, and without having to define additional libraries (*.jar files). Instead, you use the regular Shell credentials.

Deep discovery enables you to discover the topology of J2EE application systems, that is, the components of an application and not just the application itself.

This task includes the following steps:

- "Prerequisites" on page 198
- "Supported Versions" on page 198
- "Network and Protocols" on page 199
- "Discovery Workflow" on page 199
- "Discovered CITs" on page 202
- "Troubleshooting and Limitations" on page 202

1 Prerequisites

- Run **Network Discovery > Basic > Host Connection by Shell**. This job discovers hosts running NTCmd, Telnet, or SSH agents.
- Run **Host Resources and Applications by Shell**. This job discovers running software and processes relevant to JBoss.

2 Supported Versions

JBoss versions 3.x ,4.x, and 5.x.

3 Network and Protocols

- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SSH.** For credentials information, see "SSH Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

Users do not need root permissions, but do need the appropriate credentials to enable connecting to the remote machines and running the relevant commands, such as `dir\ls` and `type\cat`.

4 Discovery Workflow

Run the **J2EE Application Servers > JBoss > J2EE JBoss by Shell** job.

DFM discovers the following JBoss elements:

- ▶ **The Version Number.** DFM discovers the version number of the JBoss application server by using the following regular expression in **<JBoss base directory>\readme.html**:

```
<title>.+?\s+(.+?)\s+.
```

(That is, search for the string that follows the `<title>` string.)

If the version number is not found here, DFM discovers it by parsing the **<JBoss base directory>\<server name>\config\standardjboss.xml** file.

- ▶ **The Server Listening Port and Address.** DFM retrieves this information from the **<JBoss base directory>\server\<server name>\conf\jboss-service.xml** file.
 - ▶ The listening port is retrieved from the **RmiPort** property. The port number is needed for the JBoss Connections by JMX job to choose the relevant JMX credentials.
 - ▶ The listening address is retrieved from the **rmibindaddress** property; if this property does not exist or is set to **jboss.bind.address**, DFM uses the IP address of the Shell agent with which it connects to JBoss.

For JBoss version 5.x, DFM retrieves the listening port from the **<JBoss base directory>\server\<server name>\bootstrap\bindings.xml** file or the **<JBoss base directory>\server\<server name>\conf\bindings.xml** file.

► **The JMS Configuration.**

- DFM creates the **jboss.mq** JMS server CI according to the JBoss configuration.
- JMS destinations are parsed out from the **<JBoss base directory>\server\<server name>\deploy\jms\jbossmq-destinations-service.xml** file. For JBoss version 5.x, DFM retrieves this information from the **<JBoss base directory>\server\<server name>\deploy\messaging\destinations-service.xml** file.

► **The Database Configuration.** DFM retrieves the database configuration from the **<JBoss base directory>\server\<server name>*-ds.xml** files where

ds = data source.

There can be several of these files. By default, JBoss includes the **hsqldb-ds.xml** file which configures the OOT Hypersonic database.

► **J2EE Applications.**

DFM discovers all folders with the **.war** or **.ear** suffix under the **<JBoss base directory>\server\<server name>\tmp\deploy** directory.

For each of them, DFM finds the original **.war** or **.ear** file under the **<JBoss base directory>\server\<server name>\deploy** folder.

For each **.war** or **.ear** folder located under the **<JBoss base directory>\server\<server name>\tmp\deploy** directory, DFM creates a **J2EE Application** CI with the following attributes:

► **data_name**

For an **.ear** file, DFM retrieves the application name from the **<JBoss base directory>\server\<server name>\tmp\deploy\filename.ear\META-INF\application.xml** file.

For a **.war** file, DFM uses the original **.war** file name (under the **<JBoss base directory>\server\<server name>\deploy** folder) for the application name, but without the **.war** suffix.

➤ `j2eeapplication_isear`

Set to **true** for .ear files.

➤ `j2eeapplication_fullpath`

DFM uses the original .war file full path under the **<JBoss base directory>\server\<server name>\deploy** folder.

When discovering a JBoss server, DFM creates a **J2EE Domain** CI with the following name: **<server name>@<ipaddress>**. This action is performed also with JMX discovery.

All J2EE objects use the **J2EE Domain** CIT as a container and are deployed on a J2EE server.

➤ **Configuration Files.** DFM creates CIs for the following topology and resources configuration files:

➤ **`jboss-service.xml`** (the principal configuration file)

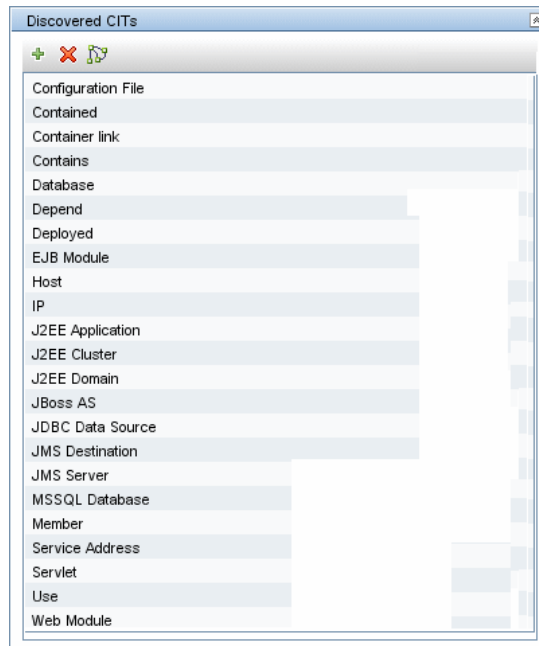
➤ the **`xxx-ds.xml`** files (for data sources)

➤ **`jbossmq-destinations-service.xml`** or **`destinations-service.xml`** for the JMS configuration

All these CIs are attached to the JBoss server CI since the JBoss server is used as a container for the J2EE Domain CI (even though DFM also creates the J2EE domain as a separate CI).

5 Discovered CITs

- The following CITs are discovered by the J2EE JBoss by Shell job:



6 Troubleshooting and Limitations

This section describes troubleshooting and limitations for JBoss discovery.

- DFM can discover a J2EE application only when its .ear file is unzipped to a folder.

19

WebLogic

This chapter includes:

Tasks

- Discover J2EE WebLogic by JMX on page 203
- Discover J2EE WebLogic by Shell on page 206

Troubleshooting and Limitations on page 210

Tasks

Discover J2EE WebLogic by JMX

This task describes how to discover WebLogic applications. The WebLogic discovery process enables you to discover a complete WebLogic topology including J2EE applications, JDBC, and JMS resources.

DFM first finds WebLogic servers based on the JMX protocol, then discovers the WebLogic J2EE environment and components.

This task includes the following steps:

- "Prerequisites" on page 204
- "Supported Versions" on page 204
- "Network and Protocols" on page 204
- "Discovery Workflow" on page 204
- "Adapter Parameters for J2EE Weblogic by JMX" on page 205
- "Discovered CITs" on page 206

1 Prerequisites

- a Set up the drivers needed to discover WebLogic. Default WebLogic drivers are included by default with the Probe installation. For details on the required *.jar files for all WebLogic versions, see "WebLogic" in *HP Universal CMDB Data Flow Management Guide*.

The *.jar files needed in discovery are located in the following folder:

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\j2ee\weblogic\<version number>.x

- b Run the **Range IPs by ICMP** job.

2 Supported Versions

The following versions are supported: WebLogic 6.x, 7.x, 8.x, 9.x, and 10.x.

3 Network and Protocols

WebLogic. For credentials information, see "WebLogic Protocol" in *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

In the Discovery Control Panel window, activate the **J2EE Weblogic Connections by JMX** job.

For details on the CIs that are discovered, see the Statistics Results table in the Details tab. For details, see "Statistics Results Pane" in *HP Universal CMDB Data Flow Management Guide*.

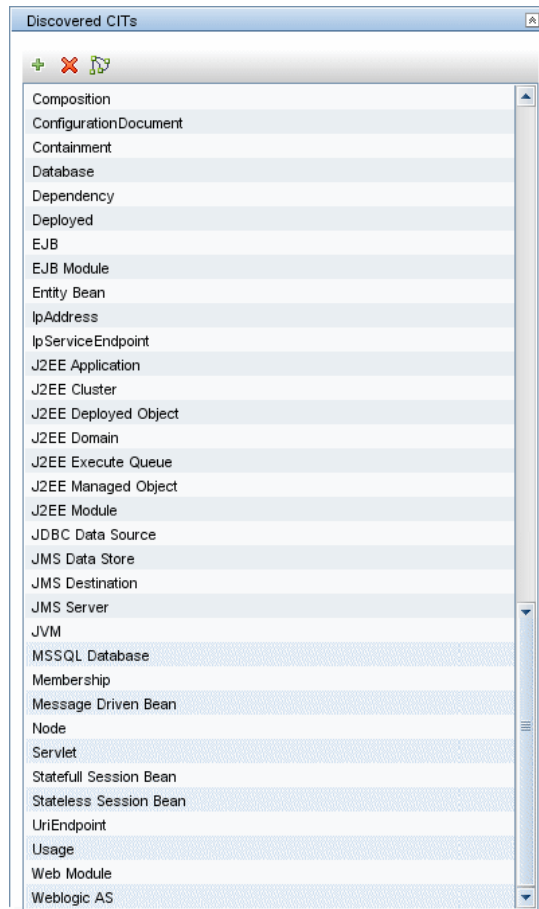
5 Adapter Parameters for J2EE Weblogic by JMX

The following adapters determine whether JMS and Application Resources related components can be omitted from reports to the UCMDB server:

- **deploymentDescriptors.** **True:** retrieves deployment descriptors of J2EE applications, EJB modules, and Web modules.
- **discoverAppResources.** **True** or missing (as previously): a full application deployment discovery is performed. **False:** only the application is discovered without all its resources.
- **discoverJMSResources.** **True** or missing (as previously): A full JMS discovery is performed. **False:** no JMS discovery is performed.

6 Discovered CITs

The following CITs are discovered by the J2EE Weblogic by JMX job:



Discover J2EE WebLogic by Shell

Note: This functionality is available as part of Content Pack 2.00 or later.

This task describes how to discover WebLogic applications by Shell and includes the following steps:

- "Prerequisites" on page 207
- "Supported Versions" on page 207
- "Network and Protocols" on page 207
- "Discovery Workflow" on page 208
- "Discovered CITs" on page 209
- "Troubleshooting and Limitations" on page 210

1 Prerequisites

- a** Run **Network Discovery > Basic > Host Connection by Shell**. This job discovers hosts running NTCmd, Telnet, or SSH agents.
- b** Run **Host Resources and Applications by Shell**. This job discovers running software and processes relevant to WebLogic.
- c** If you have created WebLogic configuration files manually, that is, you did not use the WebLogic Configuration Wizard, you must define the paths to these files:
 - Access **Adapter Management > Discovery Resources > J2EE > Adapters > WebLogic_By_Shell**.
 - In the **Adapter Definition** tab, locate the **Adapter Parameters** pane.
 - Select the **weblogic_config_root** adapter parameter.
 - Enter a comma-separated list of paths to the configuration files.

2 Supported Versions

WebLogic versions 8.x, 9.x, and 10.x.

3 Network and Protocols

- **NTCmd**. For credentials information, see "NTCMD Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- **SSH**. For credentials information, see "SSH Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

Run the **J2EE WebLogic by Shell** job.

DFM discovers the following elements:

- ▶ **The Version Number.** DFM discovers the WebLogic application server version number by retrieving it from the domain **config.xml** file.
- ▶ **The Server Listening Port and Address.** DFM retrieves this information from the domain **config.xml** file. DFM identifies the admin server in one of the following ways:
 - ▶ WebLogic version 9 or later: from the **config.xml** file, using the `<admin-server-name>` tag.
 - ▶ WebLogic version 8 or earlier: from the `<WebLogic base directory>_cfgwiz_donotdelete\startscript.xml` file, by searching for `<setenv name="SERVER_NAME">`.

If DFM does not identify the admin server, DFM stops the discovery for the current domain.

- ▶ **JMS Configurations.** DFM retrieves the JMS configuration from the domain **config.xml** file.
- ▶ **Database Configurations.** DFM retrieves the JDBC configuration from the domain **config.xml** file.
- ▶ **J2EE Applications.** DFM retrieves application names and targets from the **config.xml** file:
 - ▶ For versions earlier than WebLogic 9, the EJB and Web components are obtained from this file. Web component related information (that is, servlets and their URLs) are obtained from the `<WebLogic base directory>\<server_name>\stage\..\web.xml` files.
 - ▶ For WebLogic version 9 or later, the application EJB and Web components are retrieved from the `<WebLogic base directory>\servers\<server_name>[stage or tmp_WL_user]\...\application.xml` files.
- ▶ **Configuration Files.** DFM creates CIs for the **config.xml** (the principal configuration file).

DFM discovers WebLogic domain configurations with one of the following methods:

- ▶ DFM uses the command line to search for **platform.home** that points to the WebLogic root folder. This folder holds the domain configurations created by the WebLogic Configuration Wizard.
- ▶ DFM uses the command line to search for **-Dweblogic.RootDirectory=** which points to non-default domain configurations.
- ▶ DFM retrieves the value of the **weblogic_config_root** adapter parameter, and checks for domain configurations in all the paths specified in this parameter. For details, see "Prerequisites" on page 207.

5 Discovered CITs

The following CITs are discovered by the J2EE WebLogic by Shell job:



6 Troubleshooting and Limitations

This section describes troubleshooting and limitations for WebLogic discovery.

- ▶ DFM discovers domains only when they are created by the WebLogic Configuration Wizard.
- ▶ For versions earlier than WebLogic 9, the J2EE WebLogic by Shell job can run only on admin server hosts. For WebLogic version 9 or later, the job can run also on hosts that contain managed nodes only.
- ▶ DFM can discover a J2EE application only when its .ear file is unzipped to a folder.
- ▶ The WebLogic installation includes an example that is filtered out by default. You can remove the filter in the `weblogic_by_shell.py` Jython script. Look for `WL_EXAMPLE_DOMAINS = ['medrec']`.
- ▶ If DFM finds two domains with the same name on the same host, only one domain configuration (`j2eedomain` topology) is reported.

Troubleshooting and Limitations

WebLogic servers cannot be discovered if the WebLogic domain is configured with a domain-wide administration port. To enable discovery, access the WebLogic administrator console. In the Domain pane, clear the **Enable Administration Port** check box and save the changes.

20

WebSphere

This chapter includes:

Concepts

- ▶ WebSphere Discovery Overview on page 211

Tasks

- ▶ Discover WebSphere by JMX on page 212
- ▶ Discover WebSphere by Shell on page 214

Troubleshooting and Limitations on page 217

Concepts

WebSphere Discovery Overview

This section describes how to discover WebSphere applications. The WebSphere discovery process enables you to discover the complete WebSphere topology including J2EE applications, JDBC, and JMS resources.

Tasks

Discover WebSphere by JMX

DFM first finds WebSphere servers based on either SOAP or RMI authentication, then discovers the WebSphere J2EE environment and components.

This task describes how to discover WebSphere connections by JMX, and includes the following steps:

- "Prerequisites" on page 212
- "Supported Versions" on page 212
- "Network and Protocols" on page 213
- "Discovery Workflow" on page 213
- "Adapter Parameters for J2EE WebSphere by Shell or JMX" on page 213
- "Discovered CIs" on page 214

1 Prerequisites

- a Run the **Range IPs by ICMP** job.
- b Set up the drivers needed to discover WebSphere. Default WebSphere drivers are included by default with the Probe installation. For details on the required *.jar files, see "WebSphere" in *HP Universal CMDB Data Flow Management Guide*. - note probe installation includes WebSphere drivers for version 5 and 6, but customer can provide own drivers.

The Probe installation includes WebSphere drivers for versions 5 and 6, but you can use your own drivers, if you prefer.

The *.jar files needed in discovery are located in the following folder:

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\j2ee\websphere

2 Supported Versions

WebSphere versions 5 and 6.

3 Network and Protocols

WebSphere. For credentials information, see "WebSphere Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

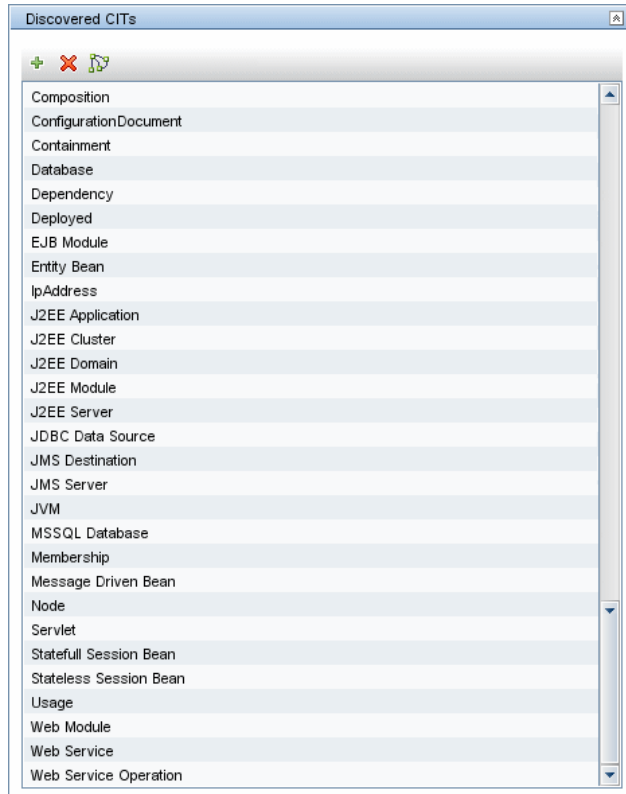
- a Run the **J2EE WebSphere by JMX** job.
- b Run the following jobs:
 - **J2EE TCP Ports**
 - **J2EE WebSphere Connections by JMX**
 - **J2EE WebSphere by Shell or JMX**

5 Adapter Parameters for J2EE WebSphere by Shell or JMX

The following adapters determine whether JMS and Application Resources related components can be omitted from reports to the UCMDB server:

- **deploymentDescriptors.** **True:** retrieves deployment descriptors of J2EE applications, EJB modules, and Web modules.
- **discoverAppResources.** **True** or missing (as previously): a full application deployment discovery is performed. **False:** only the application is discovered without all its resources.
- **discoverJMSResources.** **True** or missing (as previously): A full JMS discovery is performed. **False:** no JMS discovery is performed.
- **applications.** The list of applications that are to be discovered. The list is comma-separated.
- **discoverConfigFile.** **True:** Discovers additional configuration files for cell, server, and application.
- **discoverEAR.** **True:** Discovers J2EE application Enterprise Archive files.
- **servers.** The list of servers that are to be discovered. The list is comma-separated.

6 Discovered CIs



Discover WebSphere by Shell

Note: This functionality is available as part of Content Pack 2.00 or later.

This task describes how to discover WebSphere topology by Shell, and includes the following steps:

- "Overview" on page 215
- "Prerequisites" on page 215

- "Network and Protocols" on page 215
- "Discovery Workflow" on page 216
- "Discovered CITs" on page 217

1 Overview

This task describes how to discover WebSphere application server topology.

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 Prerequisites

- Run **Network Discovery > Basic > Host Connection by Shell**. This job discovers hosts running NTCmd, Telnet, or SSH agents.
- Run **Host Resources and Applications by Shell**. This job discovers running software and processes relevant to WebSphere.

3 Supported Versions

WebSphere versions 5.x, 6.x, and 7.x.

4 Network and Protocols

- **NTCmd**. For credentials information, see "NTCMD Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- **SSH**. For credentials information, see "SSH Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- **Telnet**. For credentials information, see "Telnet Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

Users do not need root permissions, but do need the appropriate credentials to enable connecting to the remote machines and running the relevant commands.

5 Discovery Workflow

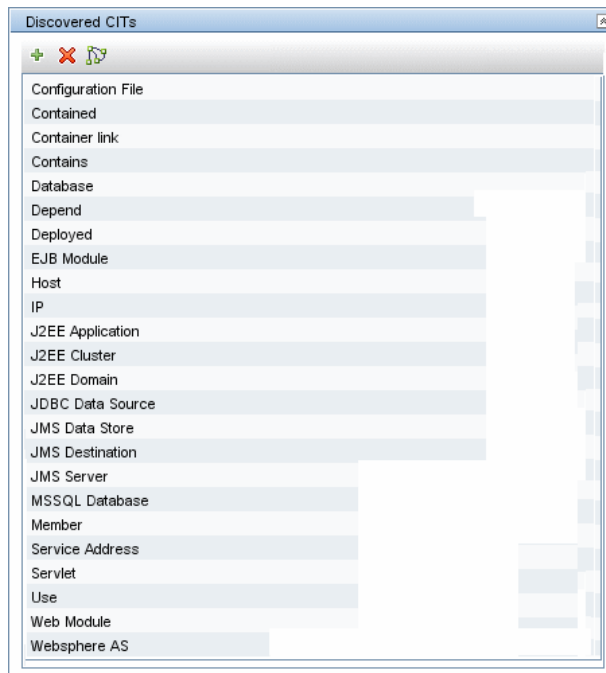
Run the **J2EE WebSphere by Shell** job.

DFM discovers the following elements:

- ▶ **The Version Number.** DFM discovers the version number of the WebSphere application server from the **WAS.product** or **BASE.product** file (depending on the WebSphere version) under the **<WebSphere base directory>\properties\version** folder.
- ▶ **The Server Listening Port and Address.** DFM retrieves information about WebSphere servers by searching for the **serverindex.xml** file, located under the **<WebSphere base directory>\profiles\<PROFILE>\config\cells\<CELL>\nodes\<NODE>** folder, or under the **<WebSphere base directory>\config\cells\<CELL>\nodes\<NODE>** folder.
- ▶ **J2EE Applications.** DFM searches for the **deployment.xml** file in each **<WebSphere base directory>\profiles\<PROFILE>\config\cells\<CELL>\applications** folder (or in the **<WebSphere base directory>\config\cells\<CELL>\nodes\<NODE>\applications** folder). The **deployment.xml** file is located in every installed application folder and contains information about application targets.
- ▶ **Configuration Files.** DFM creates CIs for the **resources.xml** resources configuration file. A CI is created for each cell, node, and server (with the relevant prefix); each CI is attached to the WebSphere server CI.

6 Discovered CITs

The following CITs are discovered by the J2EE WebSphere by Shell job:



For details on the CIs that are discovered, see the Statistics Results table in the Details tab. For details, see "Statistics Results Pane" in *HP Universal CMDB Data Flow Management Guide*.

Troubleshooting and Limitations

This section describes troubleshooting and limitations for WebSphere discovery.

Troubleshooting

Problem: A client machine includes two key stores, each one needed for identification on a specific WebSphere server. If the client attempts to connect to one of the WebSphere servers with the wrong key store, the attempt fails. If the client then uses the second, correct key store to connect to the WebSphere server, that attempt also fails.

Solution 1: Set up one key store on the client for all WebSphere servers.

Solution 2: Set up one key store per IP address range, for all WebSphere servers that use the same user name and password. For a server that uses a different user name and password, set up a key store on another IP range.

Key stores are defined in the WebSphere credentials. For details, see "WebSphere Protocol" and "Details Pane" in *HP Universal CMDB Data Flow Management Guide*.

Limitations

- ▶ If DFM finds two cells with the same name on the same host, only one cell configuration (**j2eedomain** topology) is reported.
- ▶ EJB and Web Service CIs are not discovered.
- ▶ DFM can discover a J2EE application only when its **.ear** file is unzipped to a folder.

21

Active and Passive Discovery Network Connections

This chapter includes:

Concepts

- Overview on page 219

Tasks

- Discover Processes on page 220
- Discover TCP Traffic on page 227

Concepts

Overview

All jobs in these modules run queries against the Data Flow Probe's MySQL database to retrieve data inserted by the **Host Resources and Applications** jobs. For details on Host Resource jobs, see "Host Resources and Applications Overview" on page 263.

The Data Flow Probe includes a built-in MySQL database so there is no need to install a separate MySQL instance for NetFlow. Instead, data is saved to a dedicated scheme (called **netflow** for historical reasons).

Tasks

Discover Processes

This task describes how to discover processes.

This task includes the following steps:

- "Prerequisites" on page 220
- "Job Order and Scheduling" on page 221
- "Supported Versions" on page 221
- "Discovery Pattern Parameters" on page 221
- "Network and Protocols" on page 221
- "Discovery Workflow" on page 222
- "The IP Traffic by Network Data Job" on page 224
- "The Potential Servers by Network Data Job" on page 225
- "The Server Ports by Network Data Job" on page 224
- "The Servers by Network Data Job" on page 223
- "Discovered CITs" on page 226

1 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 and Applications by Shell/SNMP/WMI** in the **Host Resources and Applications** module.

2 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 Data Flow Probe.

Although you can activate the **Host Resources and Applications** 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.

3 Supported Versions

DFM supports NetFlow versions 5 and 7.

4 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 override a process parameter value:

- a** Select **Discovery Control Panel (Advanced Mode) > Network Connections > Active Discovery**.
- b** Select the job, select the **Properties** tab, and locate the **Parameters** pane. For details on overriding a parameter, see "Parameters Pane" in *HP Universal CMDB Data Flow Management Guide*.

5 Network and Protocols

To discover network connections, define the following protocols:

- ▶ **SNMP.** For credentials information, see "SNMP Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SSH.** For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*.

Note: None of these protocols is mandatory, but WMI alone does not retrieve network data.

6 Discovery Workflow

In the Discovery Control Panel window, activate the jobs in the following order:

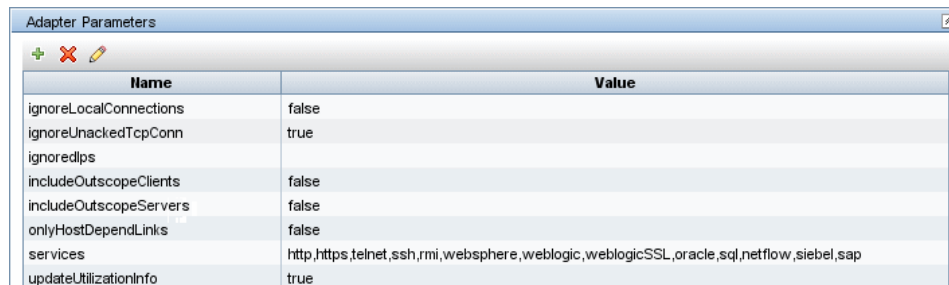
- ▶ "The Servers by Network Data Job" on page 223 (**Network Connections > Passive Discovery**)
- ▶ "The IP Traffic by Network Data Job" on page 224 (**Network Connections > Active Discovery** and **Network Connections > Passive Discovery**)
- ▶ "The Server Ports by Network Data Job" on page 224 (**Network Connections > Active Discovery**)
- ▶ "The Potential Servers by Network Data Job" on page 225 (**Network Connections > Passive Discovery**)

7 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:



Name	Value
<code>ignoreLocalConnections</code>	false
<code>ignoreUnackedTcpConn</code>	true
<code>ignoredIps</code>	
<code>includeOutscopeClients</code>	false
<code>includeOutscopeServers</code>	false
<code>onlyHostDependLinks</code>	false
<code>services</code>	http,https,telnet,ssh,rmi,websphere,weblogic,weblogicSSL,oracle,sql,netflow,siebel,sap
<code>updateUtilizationInfo</code>	true

- **`ignoreLocalConnections`**. DFM should ignore local connections. The default is **false**.
- **`ignoreUnackedTcpConn`**. DFM does not report unacknowledged connections.
- **`ignoredIps`**. 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/Servers`**. Prevents discovery of clients or servers on machines that are out of a Data Flow 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.

8 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. An attribute is defined on these links with the value of the top ports (the most important TCP/UDP ports) that are 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 ports (in which you are interested) through the maxPorts parameter.

9 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:

Name	Value
includeOutscopeServers	false
services	*

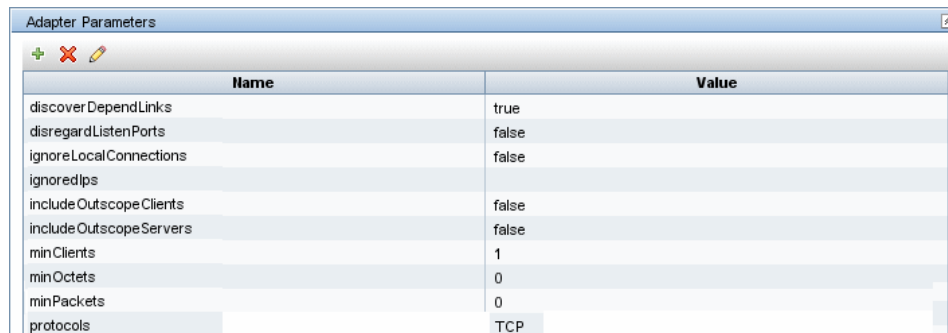
- **includeOutscopeServers.** Prevents discovery of servers on machines that are out of a Data Flow Probe's network scope.

- **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.

10 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:



Name	Value
discover DependLinks	true
disregardListenPorts	false
ignoreLocalConnections	false
ignoredIps	
includeOutscopeClients	false
includeOutscopeServers	false
minClients	1
minOctets	0
minPackets	0
protocols	TCP

- **disregardListenPorts. False:** DFM checks whether the port is marked as a listening port. If it is, DFM does not check the minimal conditions (minOctets, minClients, and minPackets). **True:** DFM does not check if the port is a listening port and instead uses the minimal conditions (minOctets, minClients, and minPackets). The default is **false**.
- **ignoreLocalConnections.** Local connections should not be discovered. The default is **false**.
- **ignoredIps.** 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/Servers.** Prevents discovery of clients or servers on machines that are out of a Data Flow Probe's network scope.

- ▶ **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**.

Caution: 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**.

11 Discovered CITs

- ▶ **Client-Server.** DFM determines which machine is the server and which the client:
 - ▶ If one end is discovered as a listening port, then this end is presumed to be a server.
 - ▶ If one end has more than two connections on its ports, it is presumed to be the server.
 - ▶ If both ends have just one connection to a port, DFM identifies whether the end is a server by checking the ports and the `portNumberToPortName.xml` file (**Adapter Management > Discovery Resources > Network > Configuration Files**).
 - ▶ If the previous is not the case, the port is checked to see whether it equals, or is less than, **1024**. In this case, DFM identifies it as a server.
- ▶ **Talk.** This link is created between two processes only if DFM does not recognize the Client-Server link between the processes. The Talk link reports bidirectionally.

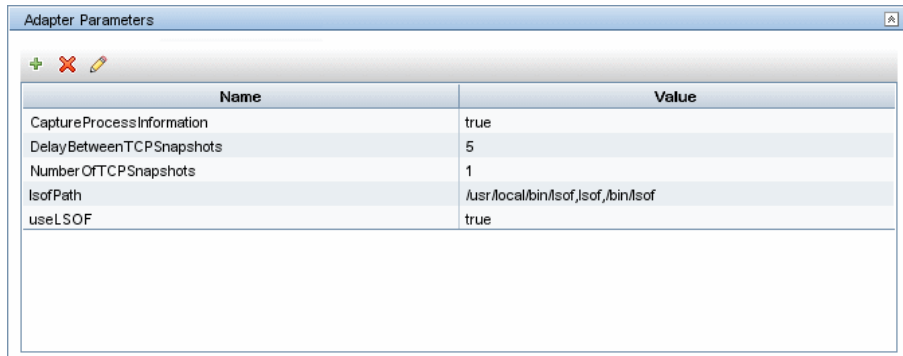
Discover TCP Traffic

Note: This functionality is available as part of Content Pack 6.00 or later.

The **TCP data by Shell** and **TCP data by SNMP** jobs enable you to collect information about TCP traffic. These jobs do not send CIs to the CMDB but run queries against existing data in the Data Flow Probe's database.

The jobs are located in the following module: (**Network Connections > Active Discovery**).

These jobs are enhanced with the following parameters that enable you to capture TCP data and to configure the time delay between captures:



Adapter Parameters	
Name	Value
CaptureProcessInformation	true
DelayBetweenTCPSnapshots	5
NumberOfTCPSnapshots	1
IsOfPath	/usr/local/bin/lssof,/bin/lssof
useLSOF	true

CaptureProcessInformation. True: process information is captured and stored in the Data Flow Probe's database. No CIs are reported. Processes are captured with the same method as that used by the Host Resources and Applications job. For details, see "Host Resources and Applications – Workflow" on page 266.

DelayBetweenTCPSnapshots. The number of seconds between TCP snapshot captures. The default is 5 seconds. It can be useful to take several TCP snapshots during a single job invocation, to retrieve more detailed data. For example, when running the **netstat -noa** command on a remote Windows system to gather TCP information, this parameter can capture process information at 5-second intervals during the command run.

NumberOfTCPSnapshots. The number of TCP snapshots to take.

IsofPath. For details, see "Adapter Parameters for the Host Resources and Applications by Shell Job" on page 268.

useLSOF. For details, see "TCP Discovery" on page 274.

22

Network Advanced

This chapter includes:

Concepts

- Overview on page 229

Tasks

- Discover DNS Zones on page 230

Concepts

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 adapter is triggered on every DNS with WMI: DFM activates the registry query to retrieve all the zones that this DNS manages.
- DFM queries each zone by activating Nslookup on the Data Flow Probe against the DNS servers.

Tasks

Discover DNS Zones

This task describes how to discover DNS zones.

This task includes the following steps:

- "Network and Protocols" on page 230
- "Discovery Workflow" on page 230
- "Discovered CITs" on page 231
- "Topology Map" on page 231
- "Troubleshooting and Limitations" on page 232

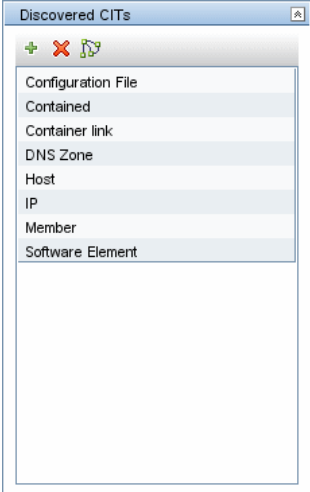
1 Network and Protocols

Nslookup (Zone transfer)

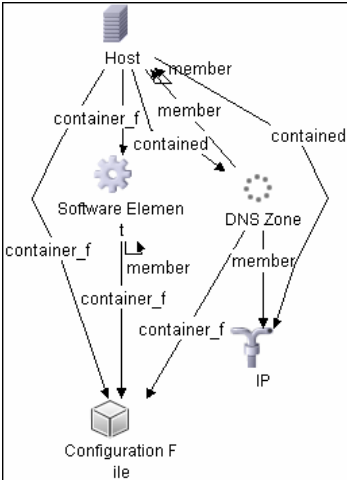
2 Discovery Workflow

In the Discovery Control Panel window, activate the **DNS Zone by Nslookup** job (Network – Advanced module).

3 Discovered CITs



4 Topology Map



5 Troubleshooting and Limitations

This section describes troubleshooting and limitations for Network discovery.

- Check that Nslookup is permitted in the environment. In many environments, Nslookup is blocked by the network administrator, in which case DFM cannot retrieve data.

23

Network Basic

This chapter includes:

Concepts

- ▶ Network – Basic Overview on page 234
- ▶ Network Workflow Overview on page 234

Tasks

- ▶ Discover Host Connection by Shell on page 235
- ▶ Host Connection by SNMP on page 237
- ▶ Discover Host Connection by WMI on page 240
- ▶ Host Connection by Shell: Discover Windows Running F-Secure on page 244

Reference

- ▶ Windows Processes on page 245
- ▶ UNIX-Based Processes on page 247

Concepts

Network – Basic Overview

You activate the jobs in the network modules to establish a Shell connection to host machines. Discovery tries to connect to the remote machine through the SSH, Telnet, and NTCmd protocols, until the first valid connection is found.

For details on using a wizard to discover the network, see "Infrastructure Wizard" in *HP Universal CMDB Data Flow Management Guide*.

Network Workflow Overview

This section describes the processes that are triggered when you activate a job in the **Network – Basic** job.

The **Network – Basic** module uses the following jobs:

- ▶ **Host Connection by Shell.** Establishes the connection to remote machines through the SSH, Telnet, and NTCMD protocols. This job discovers host type, OS information, and network connectivity information. For details, see "Discover Host Connection by Shell" on page 235.
- ▶ **Host Connection by SNMP.** Discovers SNMP agents by trying to connect to a machine using the SNMP protocol, and updates the correct host class (Windows, UNIX, router, and so on) according to the relevant OID. For details, see "Host Connection by SNMP" on page 237.
- ▶ **Host Connection by WMI.** Establishes the connection to remote machines through the WMI protocol and discovers host type, OS information, and network connectivity information. For details, see "Discover Host Connection by WMI" on page 240.

Tasks

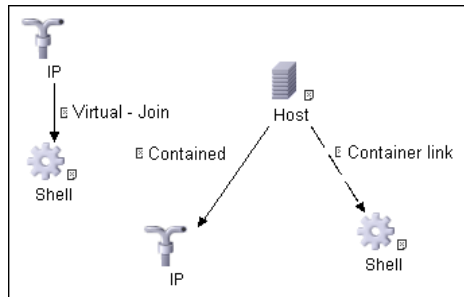
Discover Host Connection by Shell

This subject includes the following sections:

- "Input" on page 235
- "Discovery Workflow" on page 236
- "Discovered CITs" on page 237

1 Input

- **Trigger CI.** The IP address.
- **Trigger TQL.** DFM uses this query to retrieve IPs that do not have Shell or have Shell with the same IP to reconnect.



- **Node conditions.**

IP Node:

Probe Name Is NOT null
 (IP Is Broadcast Equal false OR IP Is Broadcast Is NOT null)

- **Triggered CI data.**
 - **ip_domain.** The domain of the IP address.
 - **ip_address.** The IP address itself.

- ▶ **Job parameters.**
 - ▶ **codepage.** The discovered machine codepage. Default: **NA**.
 - ▶ **language.** The discovered machine language. Default: **NA**.
 - ▶ **useAIXhwId.** Used to identify IBM AIX machines through their hardware ID. **true:** when used together with SNMP discovery, duplicate hosts may be created. **false:** no AIX LAPR is discovered. Default: **false**.
- ▶ **Protocols.**
 - ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
 - ▶ **SSH.** For credentials information, see "SSH Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
 - ▶ **Telnet.** For credentials information, see "Telnet Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

2 Discovery Workflow

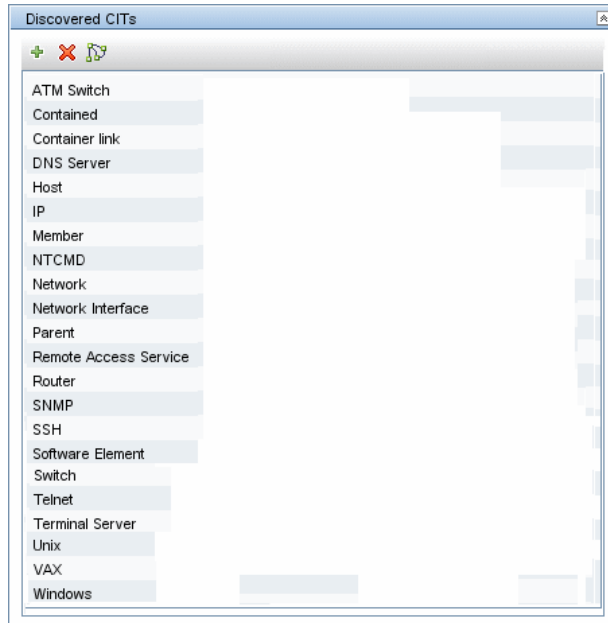
This part of the discovery depends on whether you are discovering components installed on Windows machines or UNIX-based machines. For details on the DFM processes, see:

- ▶ "Windows Processes" on page 245.
- ▶ "UNIX-Based Processes" on page 247

Note:

- ▶ DFM tries to connect using the credentials last used for this destination.
 - ▶ If the credentials do not exist, or if the connection fails, DFM tries to connect by using another protocol in a predefined list of protocols (SSH, Telnet, NTCMD) together with its credentials.
-

3 Discovered CITs



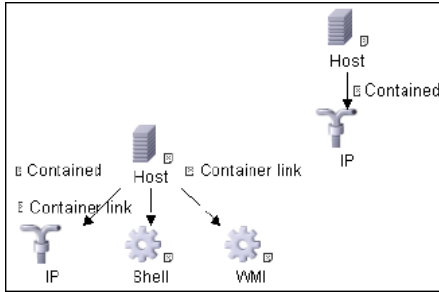
Host Connection by SNMP

This subject includes the following sections:

- "Input" on page 238
- "Discovery Workflow" on page 239
- "Discovered CITs" on page 240

1 Input

- **Trigger CI.** The IP address.
- **Trigger TQL.** This query enables the retrieval of IPs that either are not running SNMP or are running an agent with the same IP to reconnect.



➤ Node conditions.

IP Node:

```
Probe Name Is NOT null  
(IP Is Broadcast Equal false OR IP Is Broadcast Is NOT null)
```

➤ Triggered CI data.

- **ip_domain.** The domain of the IP address.
- **ip_address.** The IP address itself.
- **Job parameters.** None.

► **Protocols.**

- **SNMP.** For credentials information, see "SNMP Protocol" in *HP Universal CMDB Data Flow Management Guide*.

2 Discovery Workflow

- a** DFM runs through the credentials defined for the SNMP protocol and tries to connect successfully through one of them.
- b** DFM executes an SNMP query and obtains the class name, vendor name, host OS name, host model, host version, and host release:

Using OIDs:

SNMP MIB-2 System 1.3.6.1.2.1.1

SNMP MIB-2 Interfaces 1.3.6.1.2.1.2

The vendor's authoritative identification of the network management subsystem obtained from the system table.

- c** DFM retrieves the host IP and mask:

Using OIDs:

ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3) for subnet mask

ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4) for the least-significant bit in the IP broadcast address

ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2) for the index value which uniquely identifies the interface

- d** DFM retrieves the network interface information:

OID (1.3.6.1.2.1.2.2.1) - an interface entry containing objects at the subnetwork layer and below for a particular interface.

- e DFM retrieves the default gateway:

Used OIDs:

ipRouteDest (1.3.6.1.2.1.4.21.1.1) - for the destination IP address of this route

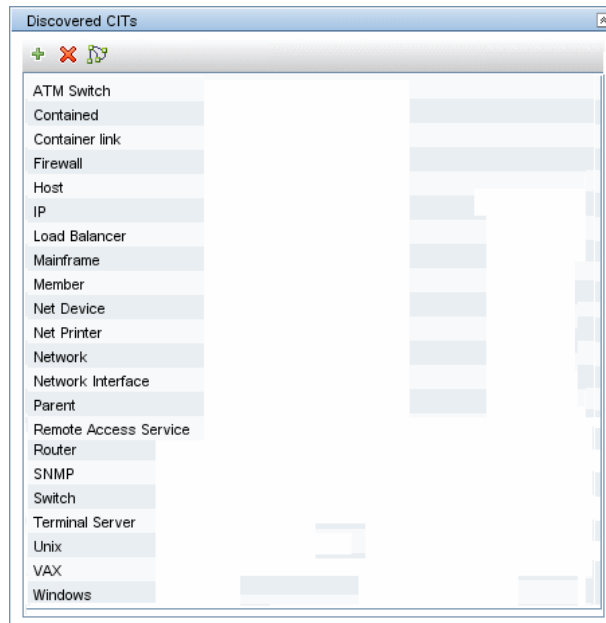
ipRouteMask (1.3.6.1.2.1.4.21.1.11) - for the mask

ipRouteDest (1.3.6.1.2.1.4.21.1.1) - for the destination IP address of this route

ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3) - for the primary routing metric for this route

ipRouteNextHop (1.3.6.1.2.1.4.21.1.7) - for the IP address of the next hop of this route

3 Discovered CITs



Discover Host Connection by WMI

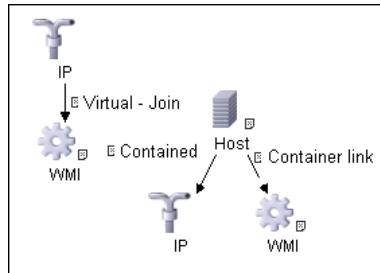
This subject includes the following sections:

- ▶ "Input" on page 241
- ▶ "Discovery Workflow" on page 241

- "Discovered CITs" on page 244

1 Input

- **Trigger CI.** The IP address.
- **Trigger TQL.** This query enables the retrieval of IPs that either are not running WMI or are running an agent with the same IP to reconnect.



- **Node conditions.**

IP Node:

Probe Name Is NOT null
(IP Is Broadcast Equal false OR IP Is Broadcast Is NOT null)

- **Triggered CI data.**
 - **ip_domain.** The domain of the IP address.
 - **ip_address.** The IP address itself.
- **Job parameters.** None.
- **Protocols.**
 - **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*.

2 Discovery Workflow

- DFM runs through the credentials defined for the WMI protocol and tries to connect successfully through one of them.
- DFM performs a WMI query for Win32_ComputerSystem to retrieve the machine name.

WMI query:

```
select Name from Win32_ComputerSystem
```

DFM performs a WMI query for Win32_NetworkAdapterConfiguration to retrieve the following interface information: IP addresses, MAC address, subnet IPs, description, and DHCP enabled attribute. DFM ignores local IPs in the interfaces.

WMI query:

```
'SELECT
DnsHostName,IPAddress,MACAddress,IPSubnet,Description,DhcpEnabled
FROM Win32_NetworkAdapterConfiguration WHERE MACAddress <> NULL'
```

- c** DFM checks whether the destination IP address is a local IP address. If it is, DFM reports IPs and hosts only.

If DFM cannot discover hosts by this manner, DFM tries to create a host defined by the lowest MAC address among the discovered network interfaces. If there is no interface to provide a valid MAC address, DFM defines the host by the destination IP address.

MAC addresses are used only in such interfaces that adhere to the following rules:

- ▶ The interface has a valid MAC address.
 - ▶ The interface does not belong to one of the following types: loopback, wireless, virtual, WAN miniport, RAS ASYNC, Bluetooth, FireWire, VPN, or IPv6 tunneling.
 - ▶ The component is not the VMware interface, and the **ignoreVmwareInterfaces** option is not set to **1** in the **globalSettings.xml** configuration file.
- d** DFM queries Win32_OperatingSystem to retrieve the host vendor, OS name, version, boot time, and installation type.

WMI query:

```
select
Caption,Version,ServicePackMajorVersion,ServicePackMinorVersion,BuildNum
ber,Organization,RegisteredUser,TotalVisibleMemorySize,LastBootUpTime,Othe
rTypeDescription from Win32_OperatingSystem
```

- e** DFM queries Win32_IP4RouteTable to retrieve the default gateway.

WMI query:

```
select NextHop, Metric1 from Win32_IP4RouteTable Where destination =
'0.0.0.0' and mask = '0.0.0.0'
```

- f** DFM queries Win32_ComputerSystem to retrieve the host manufacturer, the number of processors, host model, and OS domain.

WMI query:

```
select Manufacturer,NumberOfProcessors,Model,Domain from
Win32_ComputerSystem
```

- g** DFM retrieves the serial number by:

- ▶ Querying Win32_BaseBoard.

WMI query:

```
SELECT SerialNumber FROM Win32_BaseBoard
```

- ▶ Querying Win32_SystemEnclosure.

WMI query:

```
SELECT SerialNumber,SMBIOSAssetTag FROM Win32_SystemEnclosure
```

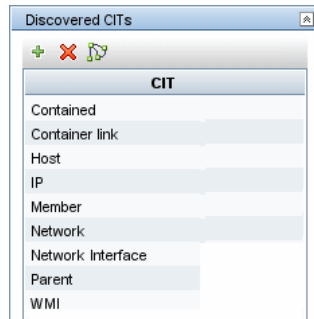
- h** DFM queries Win32_SystemEnclosure to retrieve the system asset tag.

WMI query:

```
SELECT SerialNumber,SMBIOSAssetTag FROM Win32_SystemEnclosure
```

- i If the connection is successful, DFM clears all errors and warnings that may have been generated in previous connection attempts, and returns the results.
- j If the connection is unsuccessful, DFM continues with the next WMI credential entry until all are tried.

3 Discovered CITs



Host Connection by Shell: Discover Windows Running F-Secure

When running the **Host Connection by Shell** job to discover Windows machines, on which an SSH server running the F-Secure application is installed, you must make the following modifications to F-Secure:

- ▶ Stop the F-Secure service completely.
- ▶ Verify that there are no F-Secure leftover processes still running (**fssh*** processes).
- ▶ Alter the following lines in the **sshd2_config** file. This is a F-Secure configuration file that resides in the F-Secure installation directory.
 - ▶ The **DoubleBackspace** setting should contain a **no** value, that is, DoubleBackspace no.
 - ▶ The **EmulationType** setting should contain a **raw** value, that is, EmulationType raw.

- The **EmulationTypeForCommands** setting should contain a **raw** value, that is, `EmulationTypeForCommands raw`.
- Save the altered **sshd2_config** file.
- Restart the F-Secure service.

Note:

- The Data Flow Probe enables an SSH-based connection to remote Windows machines only if the remote SSH server providers are **Open-SSH** or **F-Secure**.
 - For **Open-SSH** (that provides SSH servers for the Windows, UNIX, and Linux operating systems), DFM supports connections to Open-SSH only if the Open-SSH version is later than, or equal to, 3.7.1 (for any operating system).
-

Reference

Windows Processes

This section describes the part of the workflow that DFM performs for discovering components residing on Windows machines.

- 1** DFM discovers host attributes (OS name, version, build number, service pack, installation type). DFM starts by using the first instruction in the following list to discover the host attributes. If that fails, DFM continues to the next:

- a** WMIC "OS" object;

Full command:

```
'wmic os get caption, otherTypeDescription, version, buildnumber, csdversion /format:list < %SystemRoot%\win.ini'
```

- b** Windows registry;

Full query:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion  
VER command;  
%SYSTEMROOT%\system32\prodspec.ini processing
```

- 2** Define BIOS UUID (**wmic**)

Full command:

```
'wmic path win32_ComputerSystemProduct get uuid /format:list <  
%SystemRoot%\win.ini'
```

- 3** Define the default gateway (**netstat**).

Full command:

```
'netstat -r -n'
```

- 4** Define the DNS server IPs (**ipconfig**).

- 5** Define the boot date.

Full command:

```
'wmic OS Get LastBootUpTime /format:list < %SystemRoot%\win.ini'
```

- 6** Define the network interfaces. The **wmic** command is used first because it retrieves more information about the interface. If that fails, the output of the **ipconfig** command is used.

- a** Querying NICCONFIG object we get information about MAC address, IP addresses, interface description, subnet IPs, dynamic or static flag.

Full command:

```
'wmic nicconfig where "MACAddress <> NULL" get  
IPAddress,MACAddress,IPSubnet,Description,DhcpEnabled /format:list <  
%SystemRoot%\win.ini'
```

- b** IP filtering. Malformed and local IPs are ignored.

- 7 DFM checks whether the destination IP is local. If it is, DFM reports the host and IP only. If it is not local:
 - a DFM reports network interfaces apart from:
 - Interfaces that do not have a MAC address
 - Interfaces that belong to one of the following types: loopback, wireless, virtual, WAN miniport, RAS ASYNC, Bluetooth, FireWire, VPN, IPv6 tunneling.
 - The VMware interface, if `ignoreVmwareInterfaces` is set to `true` in the `globalSettings.xml` configuration file.
 - b DFM reports networks, IPs, and corresponding links.

UNIX-Based Processes

This section describes the part of the workflow that DFM performs for discovering components residing on UNIX-based machines:

- 1 DFM defines the OS. For details, see:
 - "FreeBSD" on page 248
 - "AIX" on page 249
 - "LINUX" on page 250
 - "HPUX" on page 251
 - "SunOs" on page 252
 - "VMKernel" on page 252

Full command:

```
'uname -a'
```

Note:

Before reporting the discovery, DFM makes the following verifications:

- ▶ If the destination IP is a virtual address, only the IP and host are reported.
 - ▶ In the case of the ZLinux OS, when the host model is **s390x**, the host is defined by the IP and domain name.
 - ▶ If the interface has an invalid MAC address, DFM does not report it.
-

FreeBSD

DFM discovers:

- 1** The DHCP enabled interfaces (**ps**).

Full command:

```
'ps aux | grep dhclient | grep -v grep'
```

- 2** The boot date (**uptime**).
- 3** The network interfaces (**name, MAC, IP, network mask, DHCP enabled flag**) and IPs (**ifconfig**).

Full command:

```
'ifconfig -a'
```

The host is defined by the lowest MAC address among the network interfaces.

- 4** The OS version and host model (**uname**).

Full command:

```
'uname -r'
```


for the version

```
'uname -m'
```

for the model

5 The domain name (**domainname**).

Report only filtered name:

```
'(none)', 'localdomain'
```

6 The BIOS UUID (**dmidecode**).

Full command:

```
'dmidecode | grep UUID'
```

7 The default gateway (**netstat**).

Full command:

```
'netstat -r -n'
```

AIX

Note: (table assumes linear flow, all flows in notes)

DFM discovers:

1 The DHCP enabled network interfaces (**ps**).

Full command:

```
'ps -aef | grep dhcpcd | grep -v grep'
```

2 The network interfaces (MAC address, name, description) (**lsdev**, **entstat**)

Full command:

```
'lsdev -Cc adapter -S | egrep ^ent'
```

- 3** The IPs (**ifconfig**).

Full command:

```
'ifconfig -a inet'
```

- 4** DFM defines the boot date, domain name, and default gateway in the same manner as for FreeBSD.
- 5** The model and vendor (**uname**).

Full command:

```
'uname -M'
```

- 6** The serial number (**lsattr**).
- 7** The OS version (**oslevel**).

LINUX

DFM discovers:

- 1** The DHCP enabled network interfaces (**ps**).

Full command:

```
'ps aux | grep dhclient | grep -v grep'
```

- 2** The IPs and network interfaces (MAC address, name, description) (**ifconfig**).

Full command:

```
'ifconfig -a'
```

- 3** The boot date, serial number (**dmidecode**), OS version, host model, domain name, and default gateway.

- 4** Information about HMC (Hardware Management Console) and its IPs (**lshmc**).

Full command:

```
'lshmc -V'
```

- 5** The BIOS UUID (**dmidecode**).

Full command:

```
'dmidecode | grep UUID'
```

- 6** The OS flavor (**redhat-release**).

Full command:

```
'cat /etc/redhat-release'
```

HPUX

- 1** DFM discovers the network interfaces by one of the following methods:

- **nwmgr**
- **lanscan** (if **nwmgr** is unsuccessful)

- 2** DFM defines aliases (**netstat**) for the discovered interfaces.

Full command:

```
'netstat -l'
```

- 3** For each interface, DFM defines IPs (**ifconfig**).

- 4** DFM discovers the host model, boot date, OS version, serial number, and default gateway.

- 5** DFM discovers the OS flavor (**swlist**).

Full command:

```
'swlist | grep -E "HPUX.*?OE"'
```

SunOs

DFM discovers:

- 1 The network interfaces (**netstat**)

Full command:

```
'netstat -np'
```

- 2 The IP addresses.

Full command:

```
'ifconfig -a'
```

- 3 The boot date, domain name, BIOS UUID, and default gateway.

- 4 The OS version and release (**uname**).

Full command:

```
'uname -rv'
```

- 5 The host model (**prtdiag**)

- 6 The manufacturer (**showrev**)

- 7 The serial number (**dmidecode**)

Full command:

```
'dmidecode | grep UUID'
```

VMKernel

DFM discovers:

- 1 The network interfaces (MAC address, name) and IPs (**esxcfg-vmknic**)

Full command:

```
'esxcfg-vmknic -l'
```

- 2 The boot date, OS version, and host model.

3 The domain name (**esxcfg-info**).

Full command:

```
'esxcfg-info | grep Domain'
```

4 The BIOS UUID (**esxcfg-info**).

Full command:

```
'esxcfg-info | grep 'BIOS UUID''
```

5 The serial number (**esxcfg-info**).

Full command:

```
'esxcfg-info -w | grep 'Serial Number''
```

6 The default gateway (**esxcfg-route**).

7 The OS flavor (**vmware**)

Full command:

```
'vmware -v'
```


24

Credential-less

This chapter includes:

Concepts

- ▶ Overview on page 255

Tasks

- ▶ Discover Host Fingerprint with Nmap on page 256

Concepts

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.

Tasks

Discover Host Fingerprint with Nmap

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:

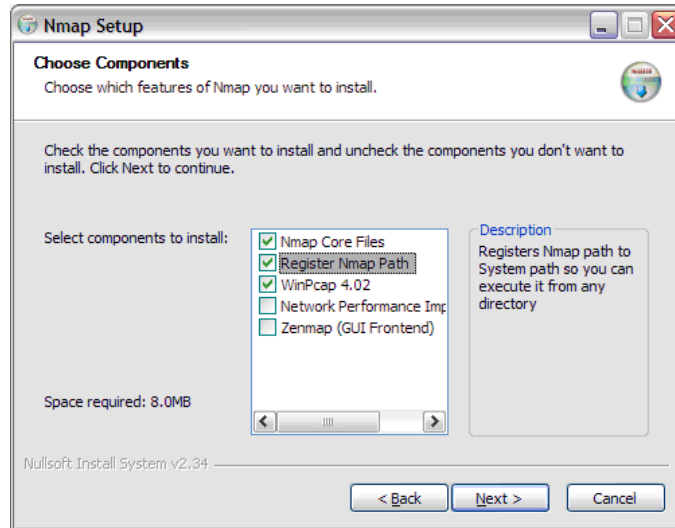
- "Prerequisites" on page 256
- "Discovery Workflow" on page 259
- "Adapter Parameters" on page 260
- "Discovered CITs" on page 261
- "Troubleshooting and Limitations" on page 261

1 Prerequisites

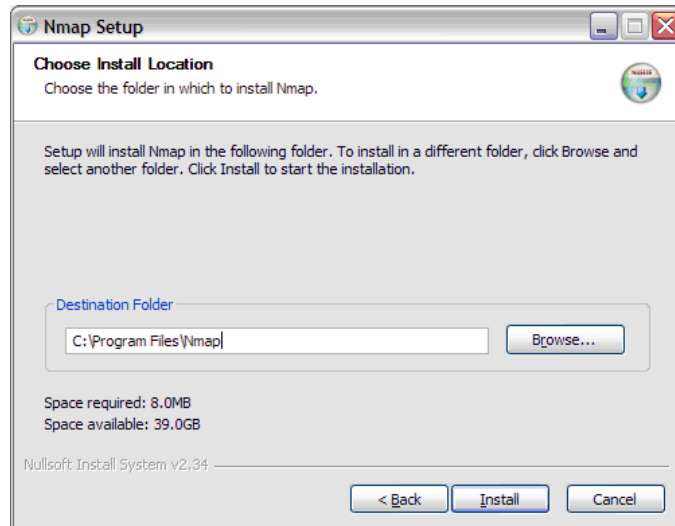
Perform the following procedure on every Data Flow Probe machine that is to run the Host Fingerprint using nmap job:

- a** Run **nmap-4.76-setup.exe** from **C:\hp\UCMDB\DataFlowProbe\tools**.
- 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.



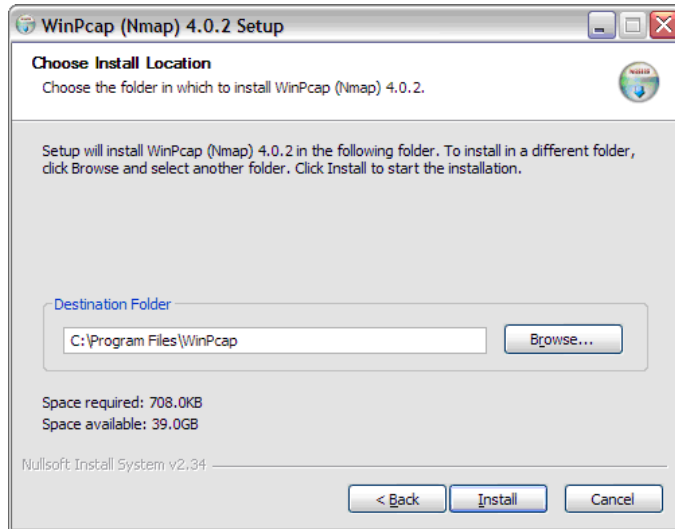
Click **Next**. The **Choose Install Location** dialog box opens.



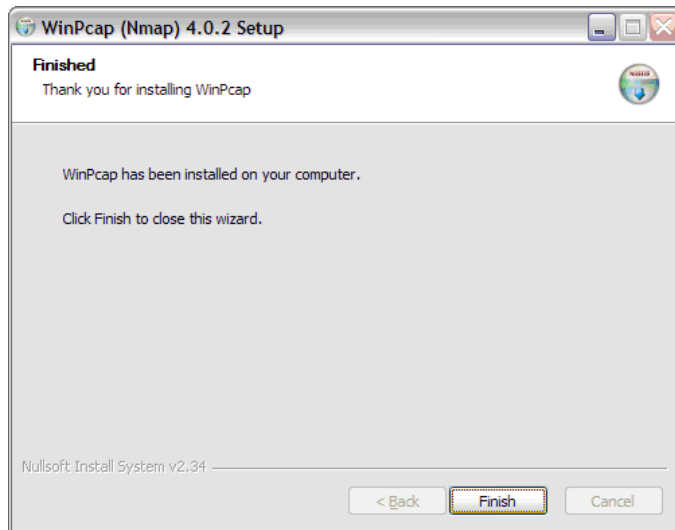
- d** Accept the default location or enter another location. Click **Install**.

Nmap is installed. The WinPcap installation dialog box opens immediately after the Nmap installation is complete.

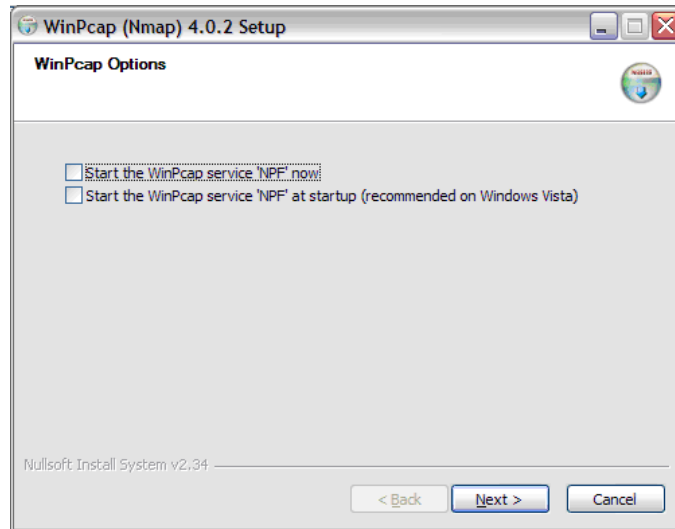
- e Accept the terms of the license and click **Next**. The **Choose Install Location** dialog box opens.



- f Accept the default location or enter another location. Click **Install**. The Finished dialog box opens.



Click **Finish**. The WinPcap Options dialog box opens.



g Clear the check boxes and click **Next**.

h Click **Finish**.

The following software is added to the Data Flow Probe machine:

- Nmap 4.76
- winpcap-nmap 4.02
- Microsoft Visual C++ Redistributable - x86 9.0.21022

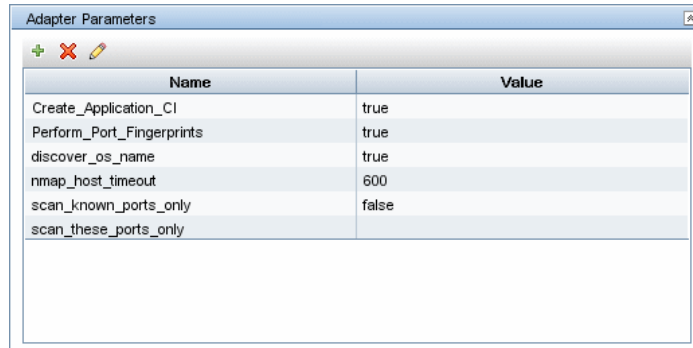
To verify, access the **Add/Remove Programs** window.

2 Discovery Workflow

This job is triggered on any discovered IP address.

3 Adapter Parameters

To view the adapter parameters: **Discovery Control Panel > Network Discovery > Credentialless Discovery > Host Fingerprint using nmap > Properties tab > Parameters pane**. For details on overriding parameters, see "Parameters Pane" in *HP Universal CMDB Data Flow Management Guide*.



Name	Value
Create_Application_CI	true
Perform_Port_Fingerprints	true
discover_os_name	true
nmap_host_timeout	600
scan_known_ports_only	false
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.
- **discover_os_name**. Discovers host OS, which may have some inaccuracy.
- **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.

4 Discovered CITs



5 Troubleshooting and Limitations

This section describes troubleshooting and limitations for Credential-less discovery.

Error Message	Reason	Solution
Can't parse XML document with Nmap results. Skipped.	nmap.exe failed before it could create a valid XML file.	<ul style="list-style-type: none"> ▶ 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.	<ul style="list-style-type: none"> ▶ Try to restart the Nmap job. ▶ Try to reduce the number of threads for the Nmap job.

Error Message	Reason	Solution
<p>The system cannot execute the specified program (in the communication log file)</p>	<p>The Windows system cannot launch the Nmap application.</p>	<p>Verify that:</p> <ul style="list-style-type: none"> ▶ The correct Nmap version has been downloaded and installed. ▶ WinPcap has been installed. <p>For details on these installations, see "Prerequisites" on page 256.</p> <p>If you have installed Nmap and WinPcap, and the error message still appears in the communication log, install vcredist_x86.exe from C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources.</p>
<p>Nmap is not installed on Probe machine</p>	<p>Nmap is not installed on the Probe machine.</p>	<p>Try to launch Nmap from the command line. Make sure that Nmap is installed. For details on the installation, see "Prerequisites" on page 256.</p>

25

Host Resources and Applications

This chapter includes:

Concepts

- ▶ Host Resources and Applications Overview on page 263

Tasks

- ▶ Host Resources and Applications – Workflow on page 266
- ▶ Revert to Previous Method of Discovering Installed Software on page 274

Troubleshooting and Limitations on page 275

Concepts

Host Resources and Applications Overview

The **Network – Host Resources and Applications** module discovers resources that exist on a host (for example, Disk, CPU, Users) as well as applications that run on that host. The module also discovers the relationships between the application and the relevant processes, the appropriate services, and the relevant IP Service Endpoint (port).

The **Host Resources and Applications by Shell/SNMP/WMI** jobs:

- ▶ Discover the TCP connections of the discovered machines, using Shell or SNMP.
- ▶ Store the information in the Data Flow Probe-dedicated netflow database.
- ▶ Query the Data Flow Probe database for TCP information.

The **Host Resources and Applications by Shell** job also gathers connectivity information (either by running **netstat** commands or the **lsof** command).

The relationships between processes and the relevant IP Service Endpoint (server port) can be discovered 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.

Note: Process to process (**P2P**) discovery is the name given to the discovery of processes running on hosts in the environment.

For details on changes made to these modules in version 8.0x, see "Changes to DDM Modules" in the *HP Universal CMDB Deployment Guide* PDF.

Job 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 the box empty, the Data Flow 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`.
 - The jobs in the **Network – Host Resources and Applications** module require a permanent connection to the Data Flow Probe's internal database. Therefore, these jobs are limited to a maximum number of 20 concurrent threads (which is the maximum number of concurrent connections permitted to the internal database).
 - For details on the Max. Threads field, see "Execution Options Pane" in *HP Universal CMDB Data Flow Management Guide*.
-

Locale-Based Processes

Note: This functionality is available as part of Content Pack 6.00 or later.

Discovery detects the locale used on a remote machine by searching for known keywords, adjusting the encoding, and using the correct regular expressions and strings. However, output may include characters in more than one language, in which case the characters may become corrupted. For example, in the following graphic, the command line uses a text file with Russian file name on an English Windows machine:



To prevent character corruption, Discovery uses a **wmic** command that saves the file in UTF-16 encoding. This is controlled by the **useIntermediateFileForWmic** parameter in the **globalSettings.xml** file (**Adapter Management > AutoDiscoveryContent > Configuration Files**). **True**: the parameter is enabled. The default value is **false**.

Tasks

Host Resources and Applications – Workflow

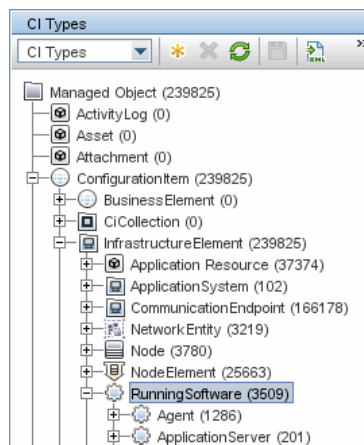
This task includes the following steps:

- "Prerequisites" on page 267
- "Network and Protocols" on page 268

- "Adapter Parameters for the Host Resources and Applications by Shell Job" on page 268
- "Adapter Parameters for the Host Resources and Applications by SNMP Job" on page 270
- "Adapter Parameters for the Host Resources and Applications by WMI Job" on page 270
- "Discovery Workflow for the Host Resources and Applications by Shell/SNMP/WMI Jobs" on page 270
- "Discovery Workflow for the Software Element CF by Shell Job" on page 271
- "TCP Discovery" on page 271
- "Discovered CITs" on page 272
- "Topology Map" on page 274

1 Prerequisites

Verify that the CMDB already contains the Agent and Shell CITs:
Modeling > CI Type Manager. Search for **RunningSoftware**, and verify that Agent and Shell are present:



2 Network and Protocols

To run this module, define the following protocols:

- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SNMP.** For credentials information, see "SNMP Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SSH.** For credentials information, see "SSH Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in the *HP Universal CMDB Data Flow Management Guide*.
- ▶ **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*.

Users do not need root permissions, but do need the appropriate credentials to enable connecting to the remote machines and running the relevant commands.

3 Adapter Parameters for the Host Resources and Applications by Shell Job

For details, see "Adapter Parameters Pane" in *HP Universal CMDB Data Flow Management Guide*.

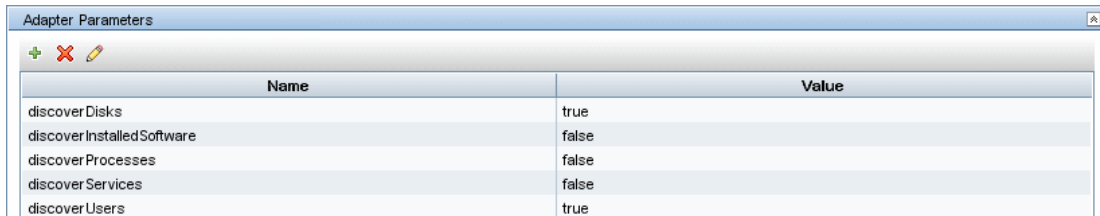


Name	Value
P2PServerPorts	*
discoverCPUs	true
discoverDisks	true
discoverInstalledSoftware	false
discoverMemory	true
discoverProcesses	false
discoverServices	false
discoverShares	true
discoverUsers	true
filterP2PProcessesByName	system,svchost.exe,lsass.exe,System Idle Process,xCmd.exe
ignoreP2PLocalConnections	false
lsOfPath	/usr/local/bin/lsOf,lsOf,/bin/lsOf
wmicPath	c:\windows\system32\wbem\

- **P2PServerPorts.** Only processes connected to these ports (as client or server) are discovered, together with this port. This parameter can include a number or a known name. You separate entries with commas. An asterisk (*) signifies all ports. The default value is *.
- **discoverProcesses. False:** Only processes that are related to specified running software are discovered. (The running software is specified in the applicationsSignature.xml file.) **True:** All processes are discovered. Previously, **False** signified that no processes are discovered.
- **discoverServices. False:** Only those services that are related to specified running software are discovered. **True:** All services are discovered.
- **discoverShares. true:** Shared resources are discovered, and **FileSystemExport** CITs are created.
- **filterP2PProcessesByName** (formerly filterProcessesByName). The names of the processes that are not reported. The default value is **system,svchost.exe,lsass.exe,System Idle Process,xCmd.exe**. To prevent P2P running, enter an asterisk (*) as the value.
- **ignoreP2PLocalConnections. False:** P2P discovery does not ignore local connections. That is, when a client and server are installed on the same host and the client-server relationship connects between them, P2P discovery should report this relationship.
- **lsofPath.** The path to the lsof command that enables process communication discovery on UNIX machines. The default value is **/usr/local/bin/lsof,lsof,/bin/lsof**.

4 Adapter Parameters for the Host Resources and Applications by SNMP Job

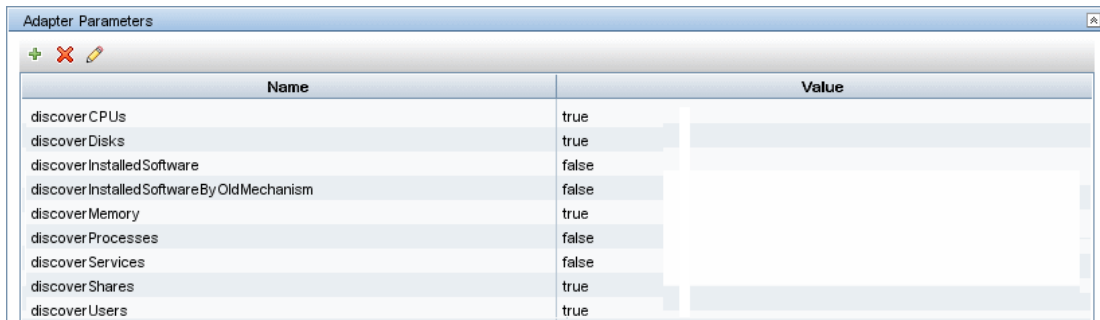
For definitions of the parameters, see "Adapter Parameters for the Host Resources and Applications by Shell Job" on page 268.



Name	Value
discoverDisks	true
discoverInstalledSoftware	false
discoverProcesses	false
discoverServices	false
discoverUsers	true

5 Adapter Parameters for the Host Resources and Applications by WMI Job

For definitions of the parameters, see "Adapter Parameters for the Host Resources and Applications by Shell Job" on page 268.



Name	Value
discoverCPUs	true
discoverDisks	true
discoverInstalledSoftware	false
discoverInstalledSoftwareByOldMechanism	false
discoverMemory	true
discoverProcesses	false
discoverServices	false
discoverShares	true
discoverUsers	true

6 Discovery Workflow for the Host Resources and Applications by Shell/SNMP/WMI Jobs

In the Discovery Control Panel window, activate the job (**Discovery Modules > Network Discovery > Host Resources and Applications > Host Resources and Applications by Shell/SNMP/WMI**).

These jobs discover resources that exist on a host (for example, Disk, CPU, Users) as well as applications that run on that host. The jobs are scheduled to run every day.

7 Discovery Workflow for the Software Element CF by Shell Job

In the Discovery Control Panel window, activate the job (**Discovery Modules > Network Discovery > Host Resources and Applications > Software Element CF by Shell**). This job retrieves the running software's configuration file and maps the file to the correct application by referring to the applicationsSignature.xml file. The triggered CIs are running software that have Shell running on their host and that include a configuration file definition that matches the definition in the applicationsSignature.xml file.

For an example on discovering Oracle configuration files, see "Discover Running Software – Scenario" in *HP Universal CMDB Data Flow Management Guide*.

8 TCP Discovery

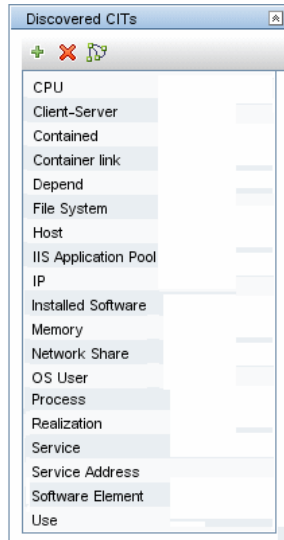
The Client/server relationship. When checking connections between two destinations (IP and port pairs), DFM uses the following logic to decide which side is the server and which the client (descending, in order of importance):

- If one of the ports is a listening port (that is, is marked as listening in the port_process table), then this port is a server port.
- If one of the ports is used by a process that is known to be a server process, then this port is the server port.
- If a local port is not listening and the remote side has not yet been processed (TCP discovery has not yet run on the remote side), it is assumed that the remote port is the server port.
- If neither port is listening and none of the processes is known to be a server process, DFM does not report P2P connectivity.

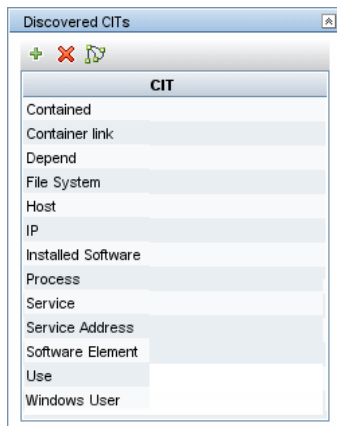
9 Discovered CITs

For details on discovered CITs, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

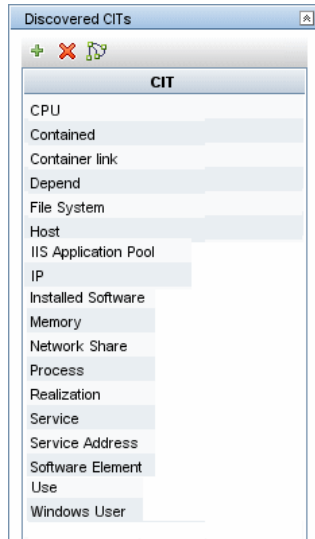
- The following CITs are discovered by the Host Resources and Applications by Shell job:



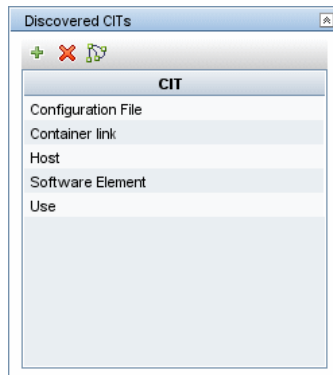
- The following CITs are discovered by the Host Resources and Applications by SNMP job:



- The following CITs are discovered by the Host Resources and Applications by WMI job:

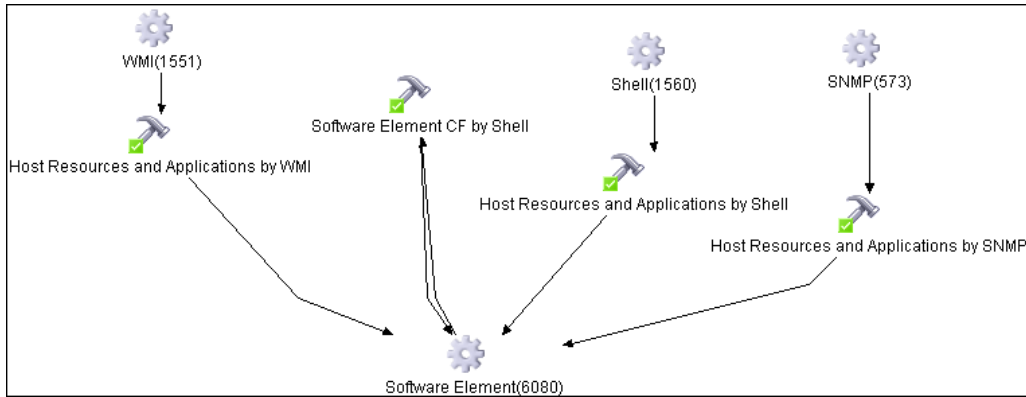


- The following CITs are discovered by the Software Element CF by Shell job:



The attributes for the resource name and its root container (that is, its parent) are updated during the run.

10 Topology Map



Revert to Previous Method of Discovering Installed Software

Note: This functionality is available as part of Content Pack 5.00 or later.

The Host Resources and Applications by WMI job discovers installed software that is installed with the WMI Windows Installer Provider. Discovery is faster than previously.

If the software is not installed with the Windows Installer, you must use the previous mechanism to discover the software.

To revert to the previous discovery mechanism for this job:

- 1** Access the Host Resources and Applications by WMI adapter: **Adapter Management > Resource Configuration > Host_Resources_By_WMI > Adapters > WMI_HR_All**.
- 2** In the **Adapter Definition** tab, locate the **Adapter Parameters** pane.
- 3** Double-click the **discoverInstalledSoftwareByOldMechanism** parameter to change the default value from **false** to **true**.

4 Save the change.

A warning message is added to the communication log.

 **Troubleshooting and Limitations**

This section describes troubleshooting and limitations for Host Resources and Applications discovery.

- To discover processes and software running on a Solaris machine, verify that the `/usr/ucb/ps` utility is installed on the Solaris machine.
- When DFM discovers installed software by WMI, and the software does not include a defined name, DFM does not report the software entity to the CMDB.

26

Layer 2

This chapter includes:

Concepts

- ▶ Overview on page 277

Tasks

- ▶ Discover Layer 2 Objects on page 278

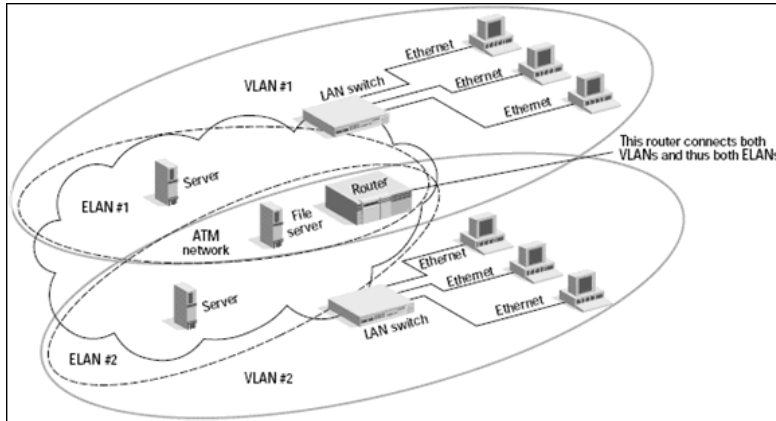
Concepts

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:



Tasks

Discover Layer 2 Objects

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:

- ▶ "Prerequisites" on page 279
- ▶ "Discovery Workflow" on page 280
- ▶ "Discovered CITs: VLANs by SNMP" on page 290
- ▶ "Discovered CITs: VLAN ports by SNMP" on page 290
- ▶ "Discovered CITs: Layer2 Topology Bridge Based by SNMP" on page 291

- "Discovered CITs: Layer2 Topology VLAN Based by SNMP" on page 291
- "Layer 2 Relationships" on page 292
- "Topology Map" on page 293
- "Troubleshooting and Limitations" on page 293

1 Prerequisites

Caution:

- All network connection jobs should finish running before you activate the Layer 2 jobs.
 - 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, DFM saves SNMP CIs to the CMDB. You should run this job on all SNMP agents on 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 adapter discovers.)

Note: Layer 2 discovery is based on the connection jobs 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.
-

2 Discovery Workflow

Caution: The Layer 2 package includes six 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** or **cisco**
- ▶ an SNMP agent that is connected to a switch that holds an operating system or model attribute value containing the string **atalyst OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% OR Host Model Like ignore case %cisco%**

**SNMP Description Like %atalyst% OR SNMP Description Like ignore
case %cisco%:**



Host Operating System Like %atalyst% OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% OR Host Model Like ignore case %cisco%:



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:

- ▶ a switch with an SNMP object that holds a description containing the string **atalyst** or **cisco**
- ▶ a switch that holds an operating system or model attribute value containing the string **atalyst OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% OR Host Model Like ignore case %cisco%**

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" in *HP Universal CMDB Data Flow Management Guide*.

SNMP Description Like %atalyst% OR SNMP Description Like ignore case %cisco%:



Host Operating System Like %atalyst% OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% OR Host Model Like ignore case %cisco%:



The `SNMP_Net_Dis_VMS_catalyst.py` script retrieves the Base MAC table and Port number If Index table.

- 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** or **cisco**
- ▶ a switch that holds an operating system or model attribute value containing the string **atalyst OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% AND Host Model Like ignore case %cisco%**

**SNMP Description Like %atalyst% OR SNMP Description Like ignore
case %cisco%:**



Host Operating System Like %atalyst% OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% AND Host Model Like ignore case %cisco%:



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. DFM 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** or **cisco**
- ▶ a switch that holds an operating system or model attribute value containing the string **atalyst OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% OR Host Model Like ignore case %cisco%**

SNMP Description Like %atalyst% OR SNMP Description Like ignore case %cisco%:



Host Operating System Like %atalyst% OR Host Model Like %atalyst% OR Host Operating System Like ignore case %cisco% OR Host Model Like ignore case %cisco%:



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.

- e** Activate the **Layer2 Enrichment** job.

This job removes Layer 2 links between physical ports and an interface that has no matching MAC address. This job is not activated automatically as part of the installation, so you should manually activate it.

- f** Activate the **Host Networking by SNMP** job.

This job discovers host networking topology using SNMP route and system tables.

3 Discovered CITs: VLANS by SNMP



4 Discovered CITs: VLAN ports by SNMP



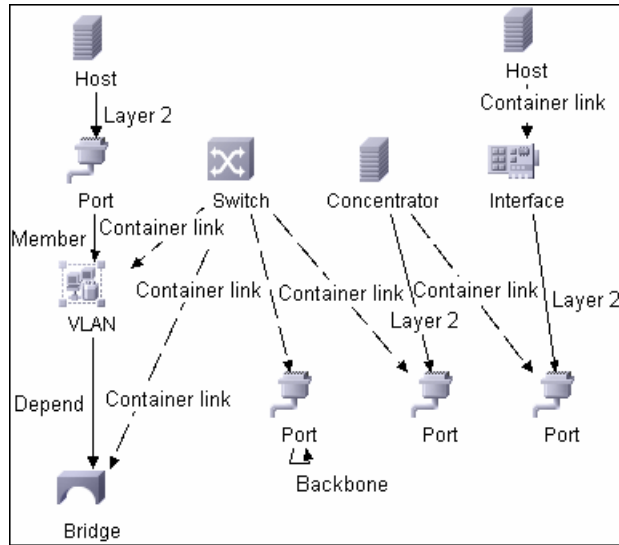
5 Discovered CITs: Layer2 Topology Bridge Based by SNMP



6 Discovered CITs: Layer2 Topology VLAN Based by SNMP

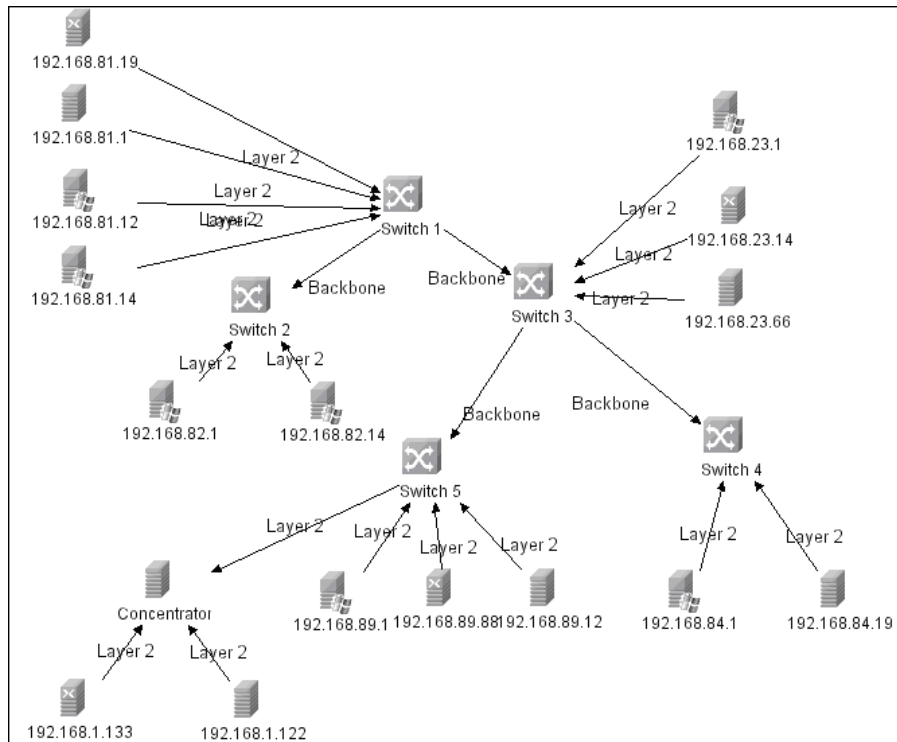


7 Layer 2 Relationships



- ▶ A Layer 2 switch can be connected to its ports directly or through a VLAN.
- ▶ The Bridge CIT 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.

8 Topology Map



9 Troubleshooting and Limitations

This section describes troubleshooting and limitations for Layer 2 discovery.

- 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 Data Flow Probe machine. For details on the MIB tables, refer to the appropriate script.
- In cases where the reported bridge MAC address is 000000000000, "", or null, the adapter 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 bridged in the destination data, the adapter returns zero results.
In the case of SNMP_Dis_L2_Bridge, bridged is set by bridge_basemacaddr.
In the case of SNMP_Dis_L2_VLAN, bridged is set by vlan_bridgemac.

27

Discovery Tools

This chapter includes:

Concepts

► Overview on page 295

Troubleshooting and Limitations on page 296

Concepts

Overview

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 28, "Importing Data from External Sources."

Troubleshooting and Limitations

This section describes troubleshooting and limitations for Siebel discovery.

Problem: The **File Monitor by Shell** does not trigger automatically. This is because there is no trigger TQL for this particular job: an automatic trigger on all destinations may cause an out-of-memory error on the Data Flow Probe.

Solution: Add the triggered CI manually.

28

Importing Data from External Sources

This chapter includes:

Concepts

- ▶ Importing Data from External Sources – Overview on page 298
- ▶ The External_source_import Package on page 300
- ▶ The Import from CSV File Job on page 301
- ▶ The Import from Properties File Job on page 305
- ▶ The Import from Database Job on page 306
- ▶ The External Source Mapping Files on page 311
- ▶ Converters on page 312

Tasks

- ▶ Import CSV Data from an External Source – Scenario on page 314

Troubleshooting and Limitations on page 318

Concepts

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 functionality to model the data and work with it. External data are mapped to CIs in the CMDB.

The following external data sources are currently supported:

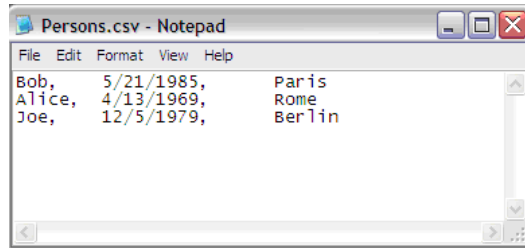
- ▶ "Comma Separated Value (CSV) Files" on page 298
- ▶ "Properties Files" on page 299
- ▶ "Databases" on page 300

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 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 DFM should start at when importing data from a CSV file:

- 1** Select **Adapter Management > Discovery Resources pane > Discovery Packages > External_source_import package > Adapters > Import_CSV.**

- 2** In the **Adapter Definition** tab, locate the **Adapter Parameters** pane.

- 3** Locate the **rowToStartIndex** parameter.

By default, the value is **1**, that is, DFM retrieves data from the first row.

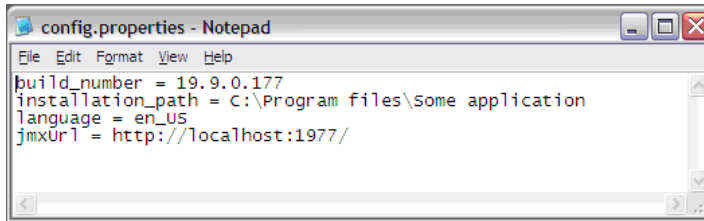
- 4** Replace **1** with the number of the row at which to start retrieving data. For example, to skip the first row and start with the second row, replace **1** with **2**.

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 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

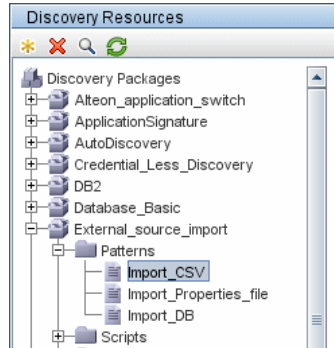
The External_source_import package consists of three jobs and three adapters. There is one job and one adapters for each external source (CSV file, properties file, database):

External Source	Job	Adapter
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 adapters are located in the **External_source_import** package:



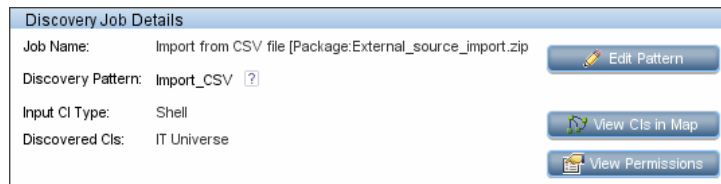
The Import from CSV File Job

This section includes the following topics:

- "Job Details" on page 301
- "Discovery Adapters Parameters" on page 302
- "Delimiters, Quotes, and Escaping Characters" on page 303

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 318.

Discovery Adapters 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 303.
- ▶ **mappingFile.** For details of the mapping file, see "The External Source Mapping Files" on page 311.
- ▶ **rowToStartIndex.** For details on setting the row at which DFM starts collecting data, see "CSV Files with Column Titles in First Row" on page 299.
- ▶ **ciType.** The CIT name. This job creates and reports CIs of this type to UCMDB, based on data in the CSV file. For example, if the CSV file contains records for UNIX hosts, you must set the **ciType** parameter to **unix**.
- ▶ **mappingString.** The string containing mapping information used to map the columns in the CSV file to the CI's attributes. You define this mapping in the following format:
 - ▶ mapping elements should be separated by commas
 - ▶ each mapping element should be specified in a **<column number>:<attribute name>** format, for example:

The string **0:host_key,1:host_hostname** defines the mapping of two attributes of a host CI, where the host's **host_key** attribute is taken from the value in the first column (0) and the **host_hostname** attribute is taken from the value in the second column (1).

For details on overriding an adapter parameter, see "Override Adapter Parameters" in *HP Universal CMDB Developer Reference Guide*.

Mapping Information for the Import from CSV File Job

You can specify mapping information for the **Import from CSV File** job with one of the following methods:

- In an external XML file. You must specify the **mappingFile** parameter. For details, see "The External Source Mapping Files" on page 311.
- Directly in a job's **ciType** and **mappingString** parameters, without using an external file.

Note: When using this mapping method, you cannot specify attribute types or converters.

If the **mappingFile** parameter is specified, the job tries to retrieve mapping information from the XML file. If it is not specified, the job uses the mapping information specified in the **ciType** and **mappingString** parameters.

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:

```
Morganfield, "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:

```
McKinley \"Muddy Waters\" Morganfield, \"April 4, 1915\"
```

This row contains two values:

- 1) McKinley "Muddy Waters" Morganfield
- 2) April 4, 1915.

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 UCMDB.

This section includes the following topics:

- "Job Details" on page 305
- "Discovery Adapter Parameters" on page 305
- "Keys and Values" on page 306
- "Comments in Properties Files" on page 306

Job Details

The job details are as follows:

Discovery Job Details	
Job Name:	Import from Properties file [Package:External_source_import.zip] Edit Pattern
Discovery Pattern:	Import_Properties_file ?
Input CI Type:	Shell View CIs in Map
Discovered CIs:	IT Universe View Permissions

This job has no Trigger TQLs associated with it.

Discovery Adapter Parameters

The following parameters are included by default:

- **ciType**. For details, see "Discovery Adapters Parameters" on page 302.
- **mappingFile**. For details of the mapping file, see "Discovery Adapters Parameters" on page 302.
- **mappingString**. For details, see "Discovery Adapters Parameters" on page 302.
- **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 an adapter parameter, see "Override Adapter Parameters" in *HP Universal CMDB Developer Reference Guide*.

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 UCMDB.

This section includes the following topics:

- "Job Details" on page 307
- "Discovery Adapter Parameters" on page 307
- "Tables and Queries" on page 308
- "Database, Schema, and Table Names" on page 308
- "Importing Data with a SQL Query" on page 309
- "Column Types" on page 309

Job Details

The job details are as follows:

Discovery Job Details	
Job Name:	Import from Database [Package:External_source_import.zip] 
Discovery Pattern:	Import_DB 
Input CI Type:	Database 
Discovered CIs:	IT Universe 

This job has no Trigger TQLs associated with it.

Discovery Adapter Parameters

The following parameters are included by default:

- **ciType**. For details, see "Discovery Adapters Parameters" on page 302.
- **mappingFile**. For details of the mapping file, see "Discovery Adapters Parameters" on page 302.
- **mappingString**. For details, see "Discovery Adapters Parameters" on page 302.
- **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 an adapter parameter, see "Override Adapter Parameters" in *HP Universal CMDB Developer Reference Guide*.

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:

Adapter Parameters	
+ ✖ ✎	
Name	
mappingFile	
schemaName	ddmi_servers
sqlQuery	
tableName	servers

The SQL query is generated from these parameters.

- Import data specifying an arbitrary SQL query as the source of the data:

Discovery Pattern Parameters	
+ ✖ ✎	
Name	
mappingFile	
schemaName	
sqlQuery	SELECT servers.* FROM servers LEFT JOIN disks C
tableName	

The SQL query is generated from the defined query. For more details, see "Importing Data with a SQL Query" on page 309.

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. DFM 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 UCMDDB. 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>
```

The External Source Mapping Files

The data in the external source is mapped to a CI's attributes in 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 adapter 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 Data Flow 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`
- ▶ `stringToByteArray`
- ▶ `stringToZippedByteArray`

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>
```

Tasks

Import CSV Data from an External Source – Scenario

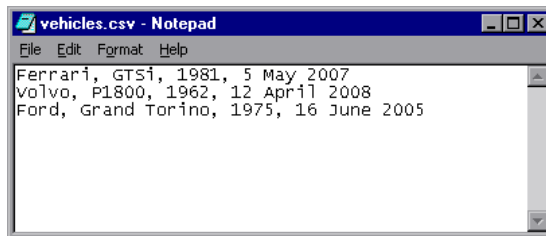
The UCMDB administrator must model a vehicle catalog that is stored in a CSV file.

This task includes the following steps:

- ▶ "Prerequisites" on page 314
- ▶ "Create a CIT" on page 315
- ▶ "Create a Mapping File" on page 316
- ▶ "Activate the Import from CSV File Job" on page 317
- ▶ "Add the Discovered Shell CI to the Job" on page 318
- ▶ "Result" on page 318

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:

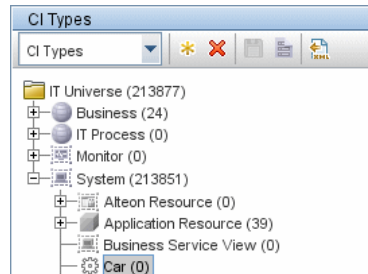
1	Name	string
2	Model	string
3	Year of manufacture	integer
4	Date of purchase	date

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 *Modeling Guide*.

- b** During the creation of the CIT, the admin adds these attributes as follows:

The screenshot shows the 'Create Configuration Item Type-Car' dialog box. It contains a table with the following attributes:

Key	Name	Display Name	Type
BODY_ICON	BODY_ICON	BODY_ICON	string
root_candidatefordel...	root_candidatefordel...	Candidate For Deleti...	date
date_of_purchase	date_of_purchase	Car Date of Purchase	date
model	model	Car Model	string
name	name	Car Name	string
year_of_manufacture	year_of_manufacture	Car Year of Manufa...	integer

For details, see "Attributes Page" in *Modeling Guide*.

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** adapter. The mapping file is located in the following folder: **Adapter Management > 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"
xsi:noNamespaceSchemaLocation=".\\mapping_schema.xsd"
parserClassName="com.hp.ucmdb.discovery.library.communication.downloader
.cfgfiles.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"
xsi:noNamespaceSchemaLocation=".\\mapping_schema.xsd"
parserClassName="com.hp.ucmdb.discovery.library.communication.downloader
.cfgfiles.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 312.

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 > Others > Discovery Tools > Import from CSV file.**

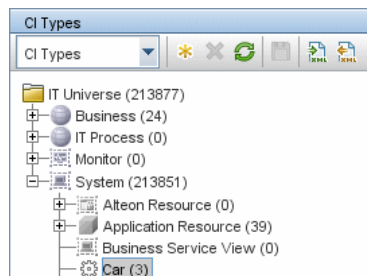
For details on activating jobs, see "Discovery Modules Pane" in *HP Universal CMDB Data Flow Management Guide*.

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" in *HP Universal CMDB Data Flow Management Guide*.

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:

- ▶ "DFM Adds Extra CI When Importing from CSV File" on page 318
- ▶ "Timeout Issues When Importing from CSV and Properties Files" on page 319

DFM 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 DFM should read the CSV file, see "CSV Files with Column Titles in First Row" on page 299.

Timeout Issues When Importing from CSV and Properties Files

Problem. When importing large CSV or properties files on the network, there may be time-out issues.

Solution. Make sure the files are not large.

29

VMware

This chapter includes:

Tasks

- Discover VMware Infrastructure Topology on page 321
- Discover VMware VMotion on page 343

Tasks

Discover VMware Infrastructure Topology

This task describes how to discover the VMware Infrastructure Topology suite of applications. You can discover virtual machines (VM), processors, memory, storage, and network resources that are running on VMware.

This task includes the following steps:

- "Supported Protocol Versions" on page 322
- "Supported VMware Servers" on page 322
- "SSL Support Details" on page 323
- "Prerequisites – Add *.jar Files" on page 323
- "Prerequisites – Run Host Discovery" on page 324
- "Prerequisites – Run WMI Discovery" on page 324
- "Prerequisites – Run Processes Discovery" on page 324
- "Prerequisites – VMware Infrastructure Permissions" on page 324

- ▶ "Network and Protocols" on page 325
- ▶ "Discovery Workflow – Overview" on page 326
- ▶ "Discovery Workflow – VMware VirtualCenter Connection by WMI and VIM" on page 326
- ▶ "Discovery Workflow – VMware VirtualCenter Topology by VIM" on page 330
- ▶ "Discovery Workflow – VMware ESX Connection by VIM" on page 334
- ▶ "Discovery Workflow – VMware ESX Topology by VIM" on page 337
- ▶ "Virtual Topology View for Non-Clusters" on page 341
- ▶ "Licensing Topology Map" on page 341
- ▶ "Troubleshooting and Limitations" on page 342

1 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 (VIM) Protocol" in *HP Universal CMDB Data Flow Management Guide*.

2 Supported VMware Servers

- ▶ vCenter Server 4.
- ▶ ESX Server 4.0. Note that DFM does not report licensing information for ESX 4.0 servers.
- ▶ ESX Server 3.5, VirtualCenter Server 2.5, and ESX Server 3i support VMware Infrastructure SDK API 2.5. The servers 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. The servers can be connected using protocol version 2.0 only.

3 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, DFM supports only one strategy (**accept all certificates always**). The following code is an example of how DFM sets the global property for the Axis engine:

```
System.setProperty('org.apache.axis.components.net.SecureSocketFactory',
    'org.apache.axis.components.net.SunFakeTrustSocketFactory')
```

4 Prerequisites – Add *.jar Files

To use the VMware Infrastructure Management protocol, add the following *.jar files from the SDK to the Data Flow 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 DFM 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\UCMDB\DataFlowProbe\content\lib directory.
- d To load the *.jar files, restart the Data Flow Probe.

5 Prerequisites – Run Host Discovery

To connect to each potential VMware server (vCenter, VirtualCenter, or ESX), discover its Host CI by running one of the **Host Connection by Shell/WMI/SNMP** jobs (in the Network – Basic module).

6 Prerequisites – Run WMI Discovery

To connect to each potential vCenter or VirtualCenter server (this is not required for ESX), make the WMI connection available for the host by running the **Host Connection by WMI** job.

7 Prerequisites – Run Processes Discovery

To connect to each potential VMware server (vCenter, VirtualCenter, or ESX), you must discover Process CIs that match certain criteria, by running one of the **Host Resources and Applications by Shell/WMI/SNMP** jobs (in the Network – Basic module).

8 Prerequisites – VMware Infrastructure Permissions

The VMware Infrastructure Management (VIM) protocol requires the following permissions:

- ▶ **System.Read** permissions for users performing discovery. Users should have permissions for all entities being discovered, and must have been assigned at least a Read-Only role.
- ▶ **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 vCenter or VirtualCenter connection adapter 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.

9 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 vCenter or VirtualCenter connectivity adapter.

The VMware Infrastructure Management (VIM) protocol is required for all VMware jobs.

- ▶ **WMI.** For credentials information, see "WMI Protocol" in *HP Universal CMDB Data Flow Management Guide*. The protocol requires the user name and password and, optionally, the domain name.
- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **SSH.** For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.

These protocols require the user name, password, and domain name (the domain name is optional for NTCmd).

- ▶ **SNMP.** For credentials information, see "SNMP Protocol" in *HP Universal CMDB Data Flow Management Guide*. This protocol requires the community name (for v2), the user name (for v3), and the password (for v3).
- ▶ **VMware Infrastructure Management (VIM).** For credentials information, see "VMware Infrastructure Management (VIM) Protocol" in *HP Universal CMDB Data Flow Management Guide*. 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.

10 Discovery Workflow – Overview

The Network – VMware module includes two jobs for vCenter or VirtualCenter Server discovery and two for ESX Server discovery:

- ▶ If the VMware Infrastructure environment is managed by vCenter or 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:

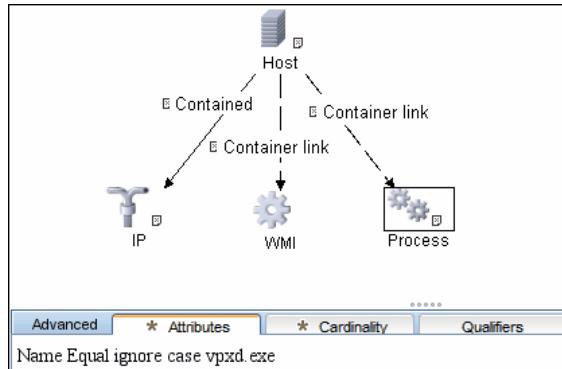
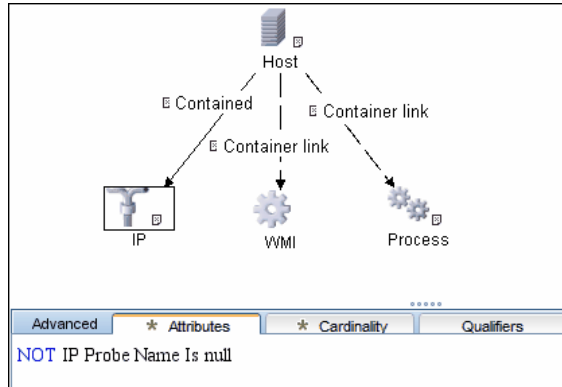
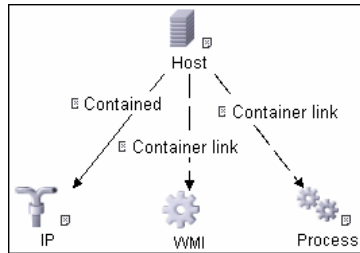
- ▶ The **Manual VMware VIM Connection** job is intended for use in those instances when the above four jobs cannot discover the VMware environment. You must, however, manually run this job, that is, you specify a URL (you need to know its format), you activate the job, and you choose the Data Flow Probe.
- ▶ DFM models the Console Operating System (COS) as a Unix CI Type, and models the hardware running the ESX as a VMWare ESX Server CI Type. Once modeled, these two CITs have the same or similar display names, but represent different entities, each one identified by its own set of unique properties.

11 Discovery Workflow – VMware VirtualCenter Connection by WMI and VIM

This job discovers vCenter or 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 Adapter Parameters. None.

DFM runs 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. DFM queries **HKLM\SOFTWARE\VMware, Inc.\VMware VirtualCenter** and searches for the **HttpsProxyPort** or **HttpProxyPort** attribute.
 - ▶ If the **HttpsProxyPort** attribute is found, DFM uses its value for the port and sets the prefix to HTTPS.
 - ▶ If the **HttpProxyPort** attribute is found, DFM uses its value for the port and sets the prefix to HTTP.

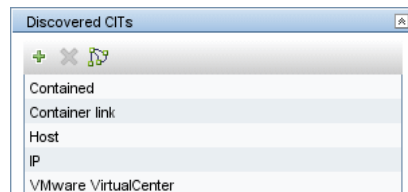
Note: DFM performs a search for the WMI port once only. The retrieved value is cached so that the same query does not need to be run for each VMware Infrastructure Management (VIM) protocol entry.

- ▶ Once the port is found, DFM generates the connection URL as follows:
<prefix>://<ip_address>:<port>/sdk.
- ▶ DFM 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, DFM tries to connect using the version 2.0 protocol.
- ▶ If the connection is successful, DFM 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.
- ▶ DFM uses the IP address to create a Host CI.

- ▶ DFM stores the generated URL used for this successful connection in the VirtualCenter CI's **connection_url** attribute.
- ▶ DFM stores the **credentialsId** of the current VIM protocol in the VirtualCenter CI's **credentialsId** attribute.
- ▶ If the connection is successful, DFM clears all errors and warnings that were generated in previous connection attempts and returns results.
- ▶ If the connection is unsuccessful, DFM continues with the next VIM protocol credentials entry, until all are tried.

Discovered CITs::



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 the user has permissions for all entities being discovered: In the **VMware Infrastructure Client**, access the **Permissions** tab of each entity (host, cluster, virtual machine, and so on). Verify that the user has been assigned at least a Read-Only role.

Note: You can view necessary permissions in the **Discovery Job Details** pane (**Discovery Control Panel > Details** tab). For details, see "Discovery Permissions Window" in *HP Universal CMDB Data Flow Management Guide*.

- **Problem.** The following error message is displayed when credentials are not correct:

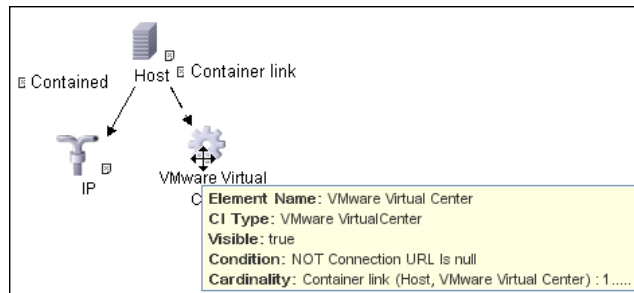
Invalid user name or password

12 Discovery Workflow – VMware VirtualCenter Topology by VIM

This job connects to vCenter or 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 vCenter or VirtualCenter Server's attribute.
- **server_url.** The URL for connecting to VMware Infrastructure, taken from the vCenter or VirtualCenter Server's **connection_url** attribute.

Discovery Adapter Parameters. **reportPoweredOffVMs.** Checks whether virtual machines that are powered off should be reported.

Data Flow Management performs the following processes:

- a** DFM extracts the connection URL and the VIM protocol credentials ID by using the VirtualCenter Trigger CI. DFM uses the credentials ID to retrieve the user name and password for the VIM protocol. DFM 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, DFM tries to connect using the version 2.0 protocol.

- b** DFM performs a query to retrieve information about Datacenters; the retrieved information is used to create Datacenter CIs.
- c** DFM 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, DFM cannot add the **licenses_total** and **licenses_available** attributes for each License Feature CI, and a warning is reported.
- d** For each Datacenter, DFM 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**). DFM does not map the **ComputeResource** resource itself to any CI (it is considered an abstract element of the hierarchy) but does use its properties.
- e** For each **ComputeResource** resource that is a **ClusterComputeResource** resource, DFM treats the resource as a cluster and creates a Cluster CI. DFM performs an additional query to retrieve its attributes.
- f** For each **ComputeResource** resource, DFM 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).
- g** For each ESX server, DFM discovers its licensing information. For details, see step c on page 331.

h When discovering VMs:

- ▶ DFM 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, DFM reports a warning and skips that VM.
- ▶ If the Tools component is installed, DFM tries to retrieve the host key. DFM searches for the lowest MAC address, or, if that is not available, for the IP. If that is also not available, DFM skips this VM and reports a warning.
- ▶ DFM determines the power status of the VM: If it is powered-off, the **reportPoweredOffVms** parameter determines whether DFM 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**, DFM tries to include the VM in the results (see the next step).

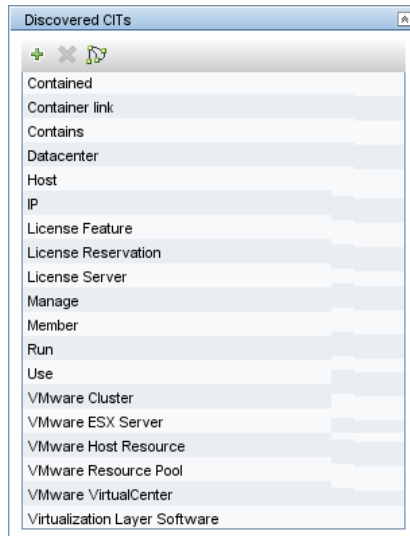
- ▶ All discovered VMs undergo a filtering mechanism. Currently filtering is performed by host keys. If there are two machines with the same host key, DFM reports only one, as follows:

If both machines are powered-on, DFM reports the first that is found.

If both machines are powered-off, DFM reports the first that is found.

If the machines have different power states, DFM reports the powered-on machine.

- i** All retrieved information is processed: DFM 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.

Discovered CITs::**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

Solution. 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 DFM 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

- **Problem.** The following warning message is displayed when DFM cannot retrieve the properties of clusters from VirtualCenter:

Failed to retrieve cluster properties, verify the connected user has sufficient permissions to query clusters information.

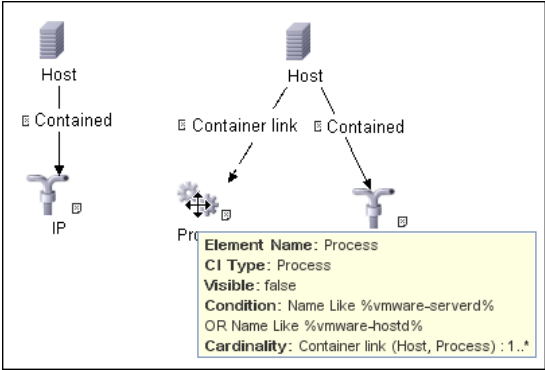
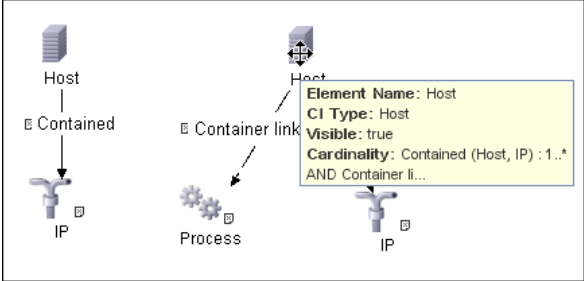
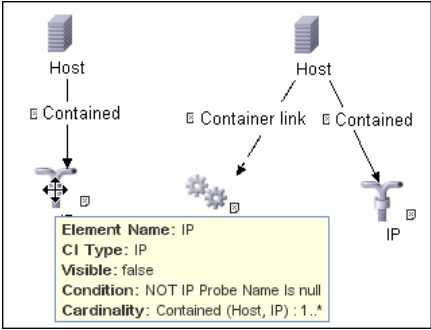
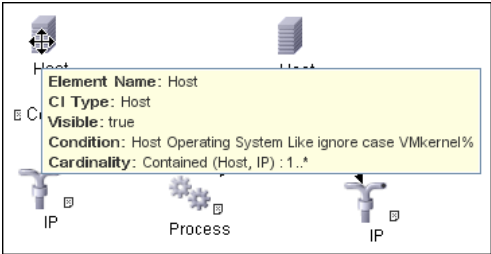
Solution. Users should have permissions for all clusters being discovered, and must have been assigned at least a Read-Only role.

13 Discovery Workflow – VMware ESX Connection by VIM

This job discovers the connections to VMware ESX servers.

Trigger CI. Unix.

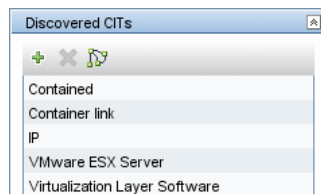
Trigger TQL and Node Conditions:



Discovery Adapter parameters. None.

Discovery and Dependency Mapping performs the following procedure:

- ▶ DFM checks the credentials for the VIM protocol.
- ▶ If the current credential includes a defined port, DFM 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.
- ▶ DFM generates a connection URL: **<prefix>://<ip_address>:<port>/sdk**.
- ▶ DFM 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, DFM tries to connect using the version 2.0 protocol.
- ▶ If the connection is successful, DFM 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, DFM 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.
- ▶ DFM clears all errors or warnings and returns all discovered results.
Otherwise, if the connection is unsuccessful, DFM tries the next VIM protocol credential, until all are tried.

Discovered CITs::**Troubleshooting and Limitations:**

- ▶ **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

Solution. 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 job completes with a time-out warning message:

<<Progress message, Severity: Error>>
VMware VIM: Timeout trying to connect to remote agent, try increasing credential timeout value

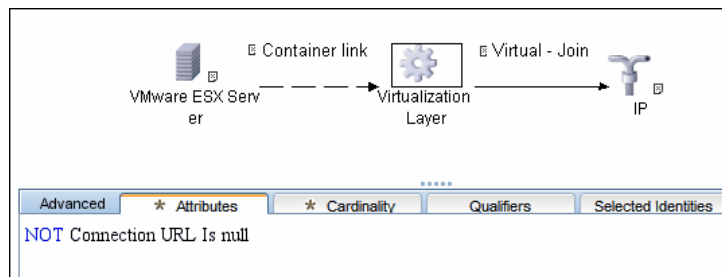
Limitation. You cannot set the connection timeout value for the job, due to VMware API limitations. The default 60 seconds timeout is always used.

14 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 Adapter Parameters. **reportPoweredOffVMs.** Checks whether VMs that are powered off should be reported.

Discovery and Dependency Mapping performs the following procedure:

- ▶ DFM 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.

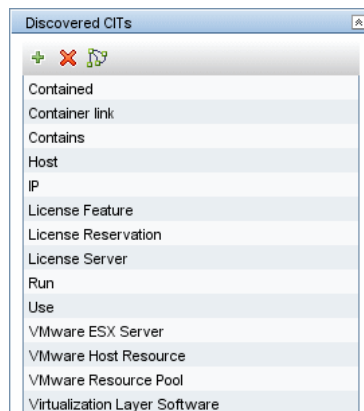
The connection is made using the version 2.5 protocol. If this connection fails, DFM tries to connect using the version 2.0 protocol.

- ▶ DFM performs discovery of the ESX servers. DFM 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 330.)

DFM discovers:

- ▶ All resource pools of the server
- ▶ All virtual machines of the server
- ▶ DFM performs discovery of the licensing information (as in the VMware VirtualCenter Topology by VIM job).
- ▶ DFM processes and returns results.

Discovered CITs:



Troubleshooting for VMware ESX Topology by VIM:

- **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>
```

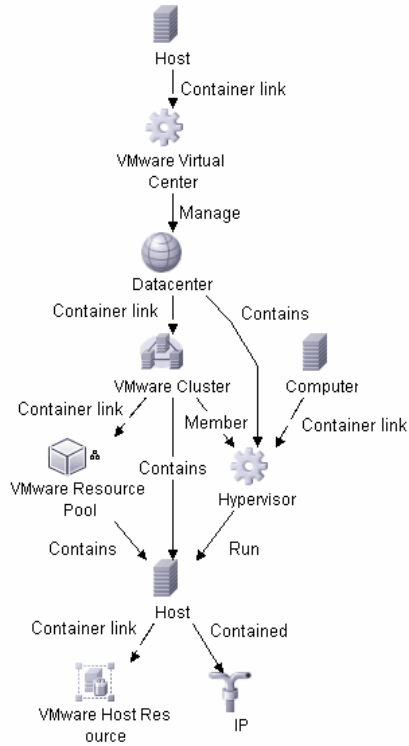
- **Problem.** The following warning message is displayed, the status is <status> and the CI is not reported:

```
Virtual machine '<vm_name>' does not have a VMware Tools running
```

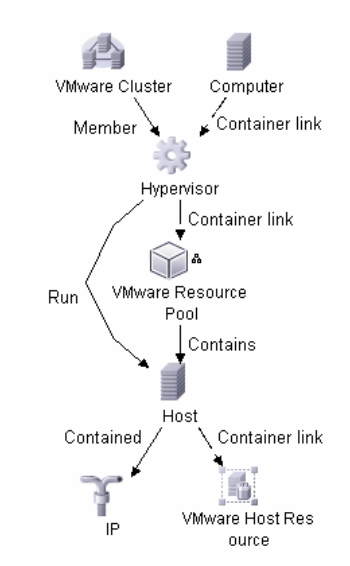
- **Problem.** The following warning message is displayed when DFM cannot retrieve license availability (permissions, in most cases, is **Global.Licenses**):

```
User does not have required '<permission>' permission, features availability information won't be reported
```

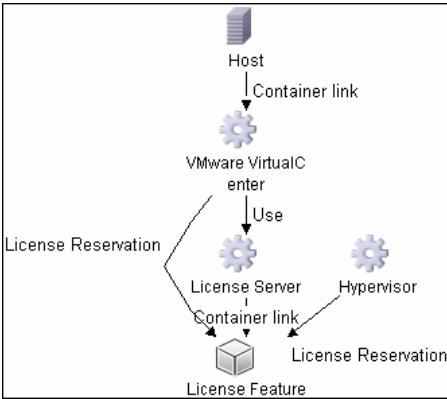
15 Virtual Topology View for Clusters



16 Virtual Topology View for Non-Clusters



17 Licensing Topology Map



18 Troubleshooting and Limitations

This section describes troubleshooting and limitations for VMware discovery.

- **Problem.** The following error message is displayed:

```
Required class %s not found. Verify VMware SDK jar files (vim.jar, vim25.jar) are present in '<PROBE>\content\lib\vmware' folder.
```

Cause. The SDK *.jar files are not copied to the Data Flow Probe.

Solution. Copy the *.jar files to the Probe, as described in "Prerequisites – Add *.jar Files" on page 323.

- **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 DFM 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" in *HP Universal CMDB Data Flow Management Guide*.
- If DFM 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.

- ▶ DFM 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, DFM 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`, `vsmmp`, `san`) that are included in the `esxFull` edition license. For these features, DFM does not report these attributes.
- ▶ There is a difference between VMware protocols 2.5 and earlier: certain attributes appear only in version 2.5 and do not appear in previous versions. As a result, when using an old protocol certain attributes are not discovered, especially for clusters and licenses.

Discover VMware VMotion

Note: This functionality is available as part of Content Pack 5.00 or later.

VMware VMotion technology moves an entire running virtual machine instantaneously from one server to another. The VMware VirtualCenter server exposes a management interface that can be used by DFM to:

- ▶ Connect to VirtualCenter using the VIM protocol, to discover its topology (Datacenters, Clusters, ESX Servers, Resource Pools, Virtual Machines, and so on).
- ▶ Connect to ESX Server and discover its full topology. This discovery is limited to the server itself.
- ▶ Listen for events that occur in the inventory structure.

VMware provides an SDK describing this interface, which includes documentation, API reference, libraries, and examples. VMware Infrastructure SDK can be downloaded from <http://www.vmware.com/support/developer/vc-sdk/>.

This task includes the following steps:

- ▶ "Supported VMware servers" on page 344
- ▶ "Prerequisites" on page 344
- ▶ "Discovery Workflow" on page 345
- ▶ "DDMDFM Package" on page 348
- ▶ "Trigger CI" on page 345
- ▶ "Trigger TQL" on page 345
- ▶ "Triggered CI Data" on page 346
- ▶ "Discovered CITs" on page 346

1 Supported VMware servers

- ▶ VI API 2.5 is supported by ESX Server 3.5, VirtualCenter Server 2.5, and ESX Server 3i (they can also be connected using protocol 2.0).
- ▶ VI API 2.0 is supported by ESX Server 3.0.x, VirtualCenter Server 2.0.x, ESX Server 3.5, VirtualCenter Server 2.5, and ESX Server 3i (protocol 2.5 is not supported).

2 Prerequisites

- a To connect to any server using the VIM protocol, prepare the following:
 - ▶ A connection URL, for example, **https://vcserver/sdk**.
 - ▶ Credentials (user name and password). A user account must be created for you on the VMware server.

For details on the protocol, see "VMware Infrastructure Management (VIM) Protocol" in Data Flow Management.

- b Permissions.** VMotion event-driven discovery requires special permissions for the protocol used:
 - ▶ **System.Read** permissions for the user performing the login, for all DFM actions. The user must be a member of the **Read-Only** user group.
- c** Discover the VMware inventory structure. For details, see "Discover VMware Infrastructure Topology" on page 321.

3 Discovery Workflow

Activate the **VMware VMotion Monitor by VIM** job. The job includes the **VMware_VMotion_discovery_by_VIM** adapter that listens for virtual machine migration events collected by the VirtualCenter server.

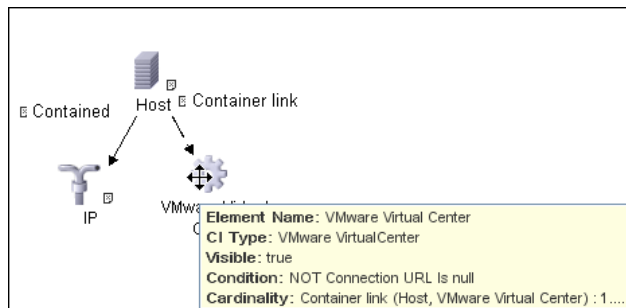
4 DFM Package

To access the **VMWare** package: **Administration > Package Manager**. For details, see "Package Manager" in *HP UCMDB Administration Guide*. For details on the contents of the package, click the link to the Readme file.

5 Trigger CI

VMware VirtualCenter

6 Trigger TQL



7 Triggered CI Data

Name	Value	Description
credentialsId	\${SOURCE.credentials_id}	The credentials ID of the VIM protocol saved in the VirtualCenter attribute.
ip_address	\${SOURCE.application_ip}	The IP address, taken from the VirtualCenter application_ip .
server_url	\${SOURCE.connection_url}	The URL for connection, taken from the VirtualCenter connection_url attribute.

8 Discovered CITs



9 Discovery Adapter Parameters

- ▶ **connectionRetryNumber**. The maximum number of times that DFM attempts to restore the connection. The default is **0** (zero), that is, the number of attempts is unlimited.
- ▶ **eventBasedDiscoveryEnabled**. If this parameter is set to **true** (the default), every time the job is activated, it stays connected to the destination machine listening for VMotion events, until the job is stopped.
- ▶ **historyHours**. The period within which DFM checks for untracked VMotion events. DFM calculates the period from when the job is activated going backwards in time. The default value is **24 hours**.

30

Apache Tomcat

Note: This functionality is available as part of Content Pack 4.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 348

Tasks

- ▶ Discover Apache Tomcat on page 349
- ▶ Discover Bugzilla, Wordpress, and MediaWiki on page 353

Concepts

Overview

To discover Apache Tomcat, DFM parses the following configuration files:

- **server.xml**. This is the main Apache Tomcat configuration file that describes the components of the Tomcat installation, its architecture, and its topology. The file also contains the configuration for global resources.

The following script fragment appears in the `server.xml` file and is the part used by the **Apache Tomcat by Shell** job to retrieve information for building the CIs:

```
<Server port="8505" shutdown="SHUTDOWN">
  <GlobalNamingResources>
    <Resource name="jdbc/GlobalDS"
      type="javax.sql.DataSource"
      driverClassName="com.inet.ora.OraDriver"
      url="jdbc:inetora:labm3mam13:1521:UCMDB" maxActive="20" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8580" protocol="HTTP/1.1"/>
    <Connector port="8509" protocol="AJP/1.3" />
    <Engine name="Catalina">
      <Host name="localhost" appBase="webapps">
        <Cluster>
          <Membership mcastAddr="228.0.0.4" mcastPort="45564"/>
        </Cluster>
      </Host>
      <Host name="grabinovic01" appBase="genadiwebapps">
        <Membership mcastAddr="228.0.0.4" mcastPort="45564"/>
      </Host>
    </Engine>
  </Service>
</Server>
```

- **context.xml**. This file defines the application context configuration. Each installed application has a unique URL prefix. This file contains resource configurations for different scopes, depending on the file location.

- **web.xml.** This file defines the application configuration, for example, the application display name and the servlets used to process HTTP requests. Currently, DFM uses this file to retrieve the application display name.

Tasks

Discover Apache Tomcat

This task describes how to discover the Apache Tomcat application.

This task includes the following steps:

- "Supported Versions" on page 349
- "Network and Protocols" on page 350
- "Discovery Workflow" on page 350
- "Apache Tomcat CITs" on page 351
- "Apache Tomcat Links" on page 351
- "Input TQL" on page 351
- "Triggered CI Data" on page 352
- "Topology View" on page 352

1 Supported Versions

Apache Tomcat versions 4.x, 5.x, and 6.x.

DFM discovers Tomcat running on the following operating systems:
Windows, UNIX, Linux.

2 Network and Protocols

Set up the following credentials:

- ▶ **NTCmd.** For credentials information, see "NTCMD Protocol" in the Data Flow Management.
- ▶ **SSH.** For credentials information, see "SSH Protocol" in the Data Flow Management.
- ▶ **Telnet.** For credentials information, see "Telnet Protocol" in the Data Flow Management.

3 Discovery Workflow

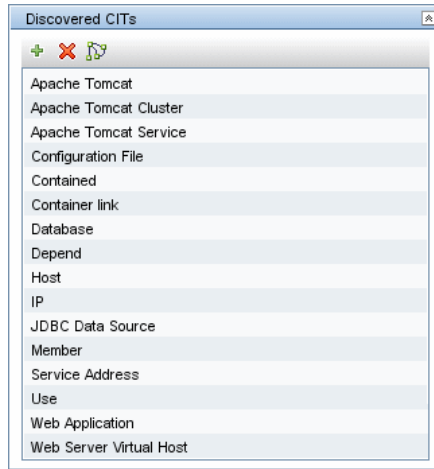
- a** Run the **Range IPs by ICMP** job (in the **Network – Basic** module) to discover IPs in the range where Tomcat is running.
- b** Run the **Host Connection by Shell** job (in the **Network – Basic** module) to discover Shell agents.
- c** Run the **Host Resources and Applications by Shell** job (in the **Network – Host Resources and Applications** module) to verify that an Apache Tomcat is running on the system, and to discover Tomcat-specific processes. If these processes are discovered, the job creates Tomcat CIs.

The job searches for the **java.exe** (or **java**) process name, then searches in the command line for either the **-Dcatalina.home=** or **-Dcatalina.base=** substring. This substring includes the path to the Tomcat home directory. If this substring is not found, the job searches for a process name starting with **tomcat** and from there acquires the path to the home directory.

The job then finds the absolute path to the Tomcat configuration file and adds this path as an attribute (**webserver_configfile**) to the Tomcat CI.

- d** Run the **Apache Tomcat by Shell** job. This job uses the Tomcat Trigger CI attribute to locate the configuration files that are discovered by the **Host Resources and Applications by Shell** job.

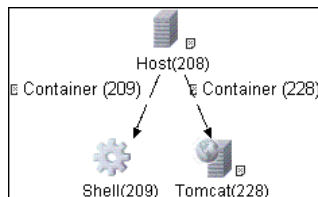
4 Apache Tomcat CITs



5 Apache Tomcat Links

- ▶ Tomcat Service > USE > Service Address
- ▶ Tomcat Service > Container > Web Server Virtual Host
- ▶ Web Application > USE > JDBC Data Source
- ▶ Tomcat > Container > Configuration File
- ▶ Tomcat > Container > Tomcat Service
- ▶ Tomcat Cluster > Member > Web Server Virtual Host
- ▶ Web Server Virtual Host > Container > Web Application
- ▶ Tomcat > Container > JDBC Data Source
- ▶ JDBC Data Source > Depend > Database

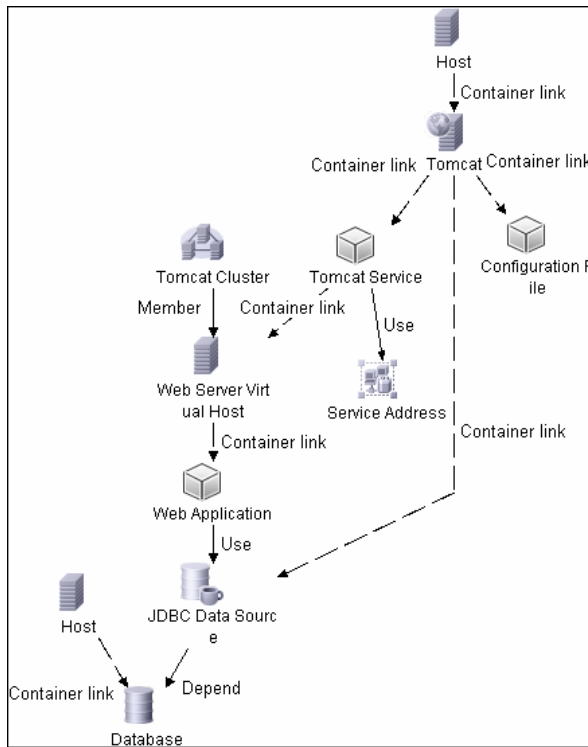
6 Input TQL



7 Triggered CI Data

Name	Value
Protocol	\${SOURCE.root_class}
configfile	\${TOMCAT.webserver_configfile}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}

8 Topology View



Discover Bugzilla, Wordpress, and MediaWiki

Note: This functionality is available as part of Content Pack 4.00 or later.

The following Web-based applications are discovered as part of the Apache and IIS discovery jobs. The following versions are supported:

Application	Supported Version
Bugzilla	3.x
Helpzilla	0.x
MediaWiki	1.15.x
Wordpress	2.5.x

To activate discovery:

- 1** Run the **Host Connection by Shell** job to create Shell CITs.
- 2** Run any of the Host Resources and Applications jobs to gather information about processes running on the host.
- 3** Run the **WebServer by Shell** job to retrieve information about Apache and available Web applications deployed on the Apache server.

The Web Application CIT:

- **ID.** webapplication
- **Parent CIT.** application
- **Usage of the existing attribute.** name
- **New attribute.** type (the type of application, for example, blog engine, wiki)

31

Microsoft Internet Information Services (IIS)

This chapter includes:

Tasks

- ▶ Discover Microsoft Internet Information Services (IIS) – Previous Topology on page 355
- ▶ Discover Microsoft Internet Information Services (IIS) – Current Topology on page 358

Tasks

Discover Microsoft Internet Information Services (IIS) – Previous Topology

This task describes how to discover Internet Information Services (IIS). IIS is a set of Internet-based services for servers created by Microsoft for use with Microsoft Windows.

This task includes the following steps:

- ▶ "Supported Versions" on page 356
- ▶ "Network and Protocols" on page 356
- ▶ "Discovery Workflow" on page 356
- ▶ "Discovered CITs" on page 356
- ▶ "Topology Map" on page 358

- ▶ "Troubleshooting and Limitations" on page 358

1 Supported Versions

IIS versions 5 and 6.

2 Network and Protocols

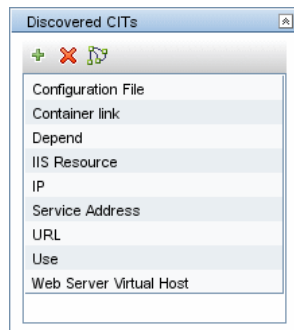
NTCmd. For credentials information, see "NTCMD Protocol" in Data Flow Management. Verify that the target machine running IIS lies in the Data Flow Probe range.

3 Discovery Workflow

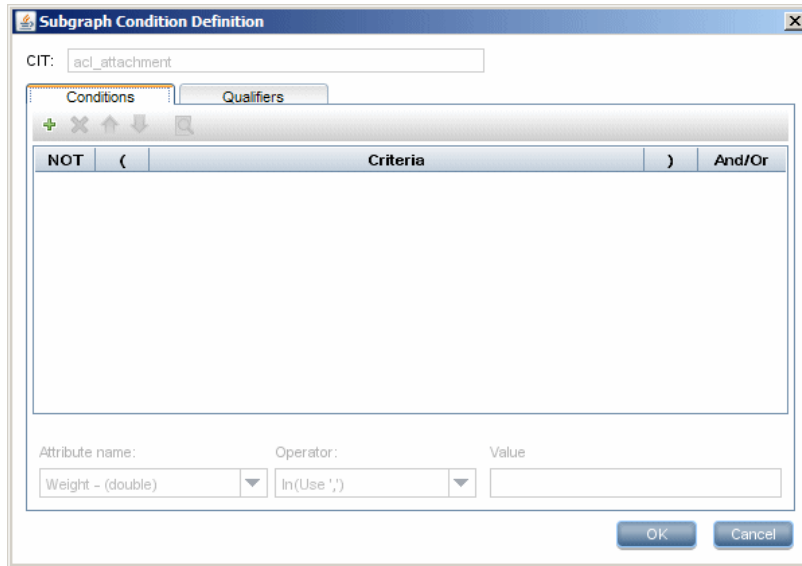
In the Discovery Control Panel window, activate the jobs in the following order:

- ▶ **WebServices by URL.** For a limitation, see "Troubleshooting and Limitations" on page 358.
- ▶ **IIS Applications by NTCMD**

4 Discovered CITs

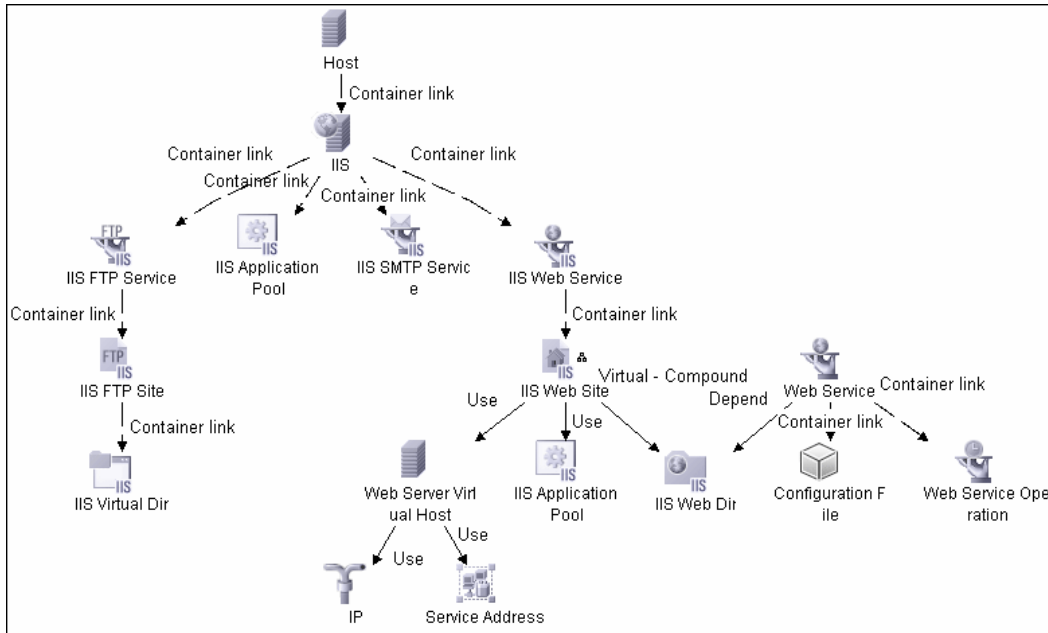


The dependency list for the IIS Web Site node is defined as follows:



For details, see "Subgraph Condition Definition Dialog Box" in *Modeling Guide*.

5 Topology Map



6 Troubleshooting and Limitations

This section describes troubleshooting and limitations for IIS discovery.

- ▶ When activating the **WebServices by URL** job, the Web services being discovered should be authentication-less, that is, they can be accessed without credentials (user name/password).

Discover Microsoft Internet Information Services (IIS) – Current Topology

Note: This functionality is available as part of Content Pack 4.00 or later.

This task describes how to discover Microsoft Internet Information Services (IIS).

This task includes the following steps:

- "Supported Versions" on page 359
- "Prerequisites" on page 359
- "Network and Protocols" on page 359
- "Discovery Workflow" on page 360
- "Trigger TQL" on page 361
- "Triggered CI Data" on page 361
- "Parameters" on page 361
- "Discovered CITs" on page 362
- "IIS Package" on page 362
- "Permissions" on page 363
- "Topology Map" on page 364
- "Troubleshooting and Limitations" on page 365

1 Supported Versions

IIS version 7.0 or earlier.

2 Prerequisites

- To retrieve all relevant information, DFM should be able to execute Visual Basic scripts and have write permission to the `%SystemRoot%/system32/drivers/etc` folder.
- Verify that the target machine running IIS lies in the Data Flow Probe range.

3 Network and Protocols

NTCmd. For credentials information, see "NTCMD Protocol" in Data Flow Management.

4 Discovery Workflow

In the Discovery Control Panel window, activate the jobs in the following order:

- a** Run the **Host Connection by Shell** job to create Shell CITs.
- b** Run the **Host Resources and Applications by Shell** job to discover IIS Web Server CIs and IIS Application Pool CIs with corresponding **Depend** links to the managing process.
- c** Run the **IIS Applications by NTCMD** job to discover the detailed topology of IIS.

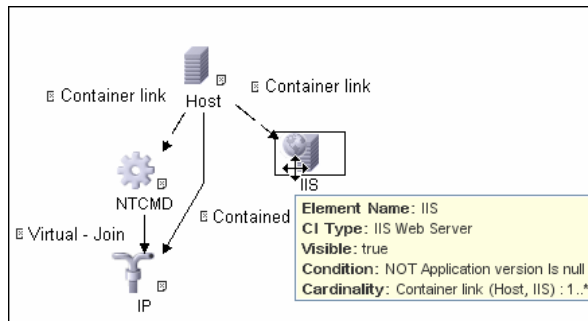
After the connection is made, DFM copies the **adsutil.vbs** script on the remote machine. DFM retrieves IIS topology information from the output of this tool.

Microsoft IIS version 7.0 enables you to create an IIS application from a Web directory, as well as from a virtual directory (as in prior versions). Therefore, when DFM discovers such an application, DFM creates an IIS Web Directory CI.

To view required permissions: **Discovery Control Panel > Advanced Mode > Web Servers > IIS > IIS Applications by NTCMD job. Details tab > Discovery Job Details** pane. Click the **View Permissions** button. For details, see "Permissions" on page 363.

Note: The IIS Web Dir CI is created only if there is an IIS Virtual Dir CI or a web.config file underneath in the topology, otherwise it is not reported.

5 Trigger TQL



6 Triggered CI Data

Name	Value
credentialsId	\${NTCMD.credentials_id}
iis_name	\${SOURCE.data_name}
iis_version	\${SOURCE.application_version_number}
ip_address	\${NTCMD.application_ip}

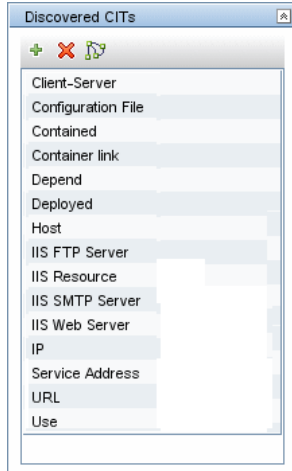
7 Parameters

Override	Name	Value
<input type="checkbox"/>	adsutil_path	adsutil.vbs
<input type="checkbox"/>	do_web_service	true
<input type="checkbox"/>	report_legacy_topology	true

- **adsutil_path.** Enter the path and name to the **adsutil.vbs** script. The adsutil.vbs script is a free script provided by Microsoft for IIS management tasks.
- **do_web_service. true:** The **IIS Web Service** CI is reported. Note that **report_legacy_topology** must also be set to **true** for DFM to report this CI.

- **report_legacy_topology. true:** For backwards compatibility, DFM continues, by default, to report the legacy IIS topology.

8 Discovered CITs



9 IIS Package

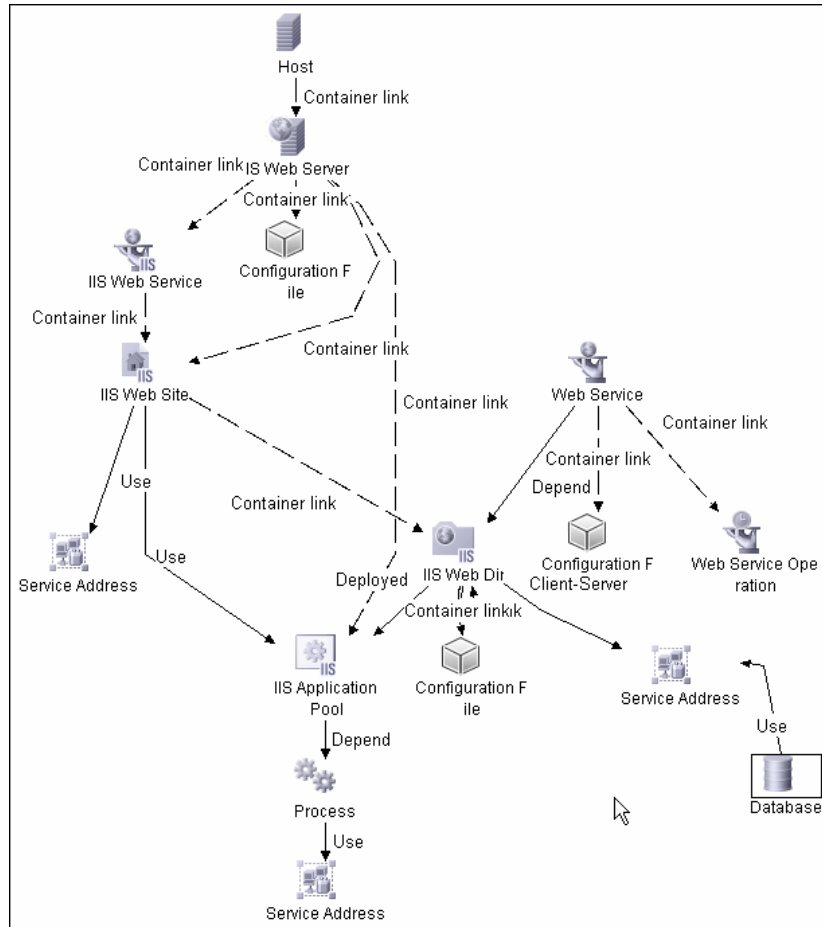
All components responsible for IIS discovery by Shell are bundled in the IIS package (in the Web Tier category).

10 Permissions

Discovery Permissions			
Required Permissions			
Permission	Operation	Usage Description	Objects and Parameters
Shell	exec	Basic login	uname ver
Shell	copy	Copy file to remote machine	adsutil.vbs - Visual Basic script for IIS discovery
Shell	exec	Discover IIS Topology	cscript.exe adsutil.vbs ENUM "MSFTPSVC/{SITENUM}root" cscript.exe adsutil.vbs ENUM "W3SVC" cscript.exe adsutil.vbs ENUM "W3SVC/AppPools" cscript.exe adsutil.vbs ENUM "W3SVC/AppPools/{POOLNAME}" cscript.exe adsutil.vbs ENUM "W3SVC/{SITENUM}" cscript.exe adsutil.vbs ENUM "W3SVC/{SITENUM}/root" cscript.exe adsutil.vbs ENUM /p MSFTPSVC cscript.exe adsutil.vbs ENUM /p MSFTPSVC/{SITENUM}/Root cscript.exe adsutil.vbs ENUM /p W3SVC cscript.exe adsutil.vbs ENUM /p W3SVC/AppPools cscript.exe adsutil.vbs ENUM MSFTPSVC cscript.exe adsutil.vbs ENUM MSFTPSVC/{SITENUM} cscript.exe adsutil.vbs ENUM SMTPSVC cscript.exe adsutil.vbs GET "{PATH}/KeyType" cscript.exe adsutil.vbs GET KeyType cscript.exe adsutil.vbs GET MSFTPSVC/{SITENUM}/Root/{PATH}/KeyT... cscript.exe adsutil.vbs GET MaxBandwidth dir /B hostname

Close

11 Topology Map



12 Bugzilla, Wordpress, and MediaWiki Discovery

Note: This functionality is available as part of Content Pack 4.00 or later.

For details see "Discover Bugzilla, Wordpress, and MediaWiki" on page 353.

13 Troubleshooting and Limitations

This section describes troubleshooting and limitations for IIS discovery.

- ▶ An IIS Web server CI is created even if no Web service is running on the machine but the IIS FTP and IIS SMTP services are present.

Part III

Supported Integrations

32

Network Node Manager *i* (NNMi) Integration with HP Universal CMDB

This chapter includes:

Concepts

- ▶ NNMi Integration Overview on page 370
- ▶ NNMi - UCMDB Integration Architecture on page 371

Tasks

- ▶ Set Up HP NNMi-HP UCMDB Integration on page 372
- ▶ Run HP NNMi-UCMDB Integration on page 373
- ▶ Use the HP NNMi-HP UCMDB Integration on page 381
- ▶ Change the HP NNMi-HP UCMDB Integration Configuration on page 384
- ▶ Disable HP NNMi-HP UCMDB Integration Configuration on page 384
- ▶ Perform Impact Analysis on page 385

Reference

- ▶ HP NNMi-HP UCMDB Integration Configuration Form Reference on page 385
- ▶ NNMi Protocol Connection Parameters on page 389

Troubleshooting and Limitations on page 390

Concepts

■ NNMi Integration Overview

You integrate NNMi with UCMDB using the Data Flow Management (DFM) application.

When you activate the **Integration – NNM Layer2** module, DFM retrieves Layer 2 network topology data from NNMi and saves the data to the Universal CMDB database. Users can then perform change management and impact analysis through the UCMDB correlation engine.

Note: DFM version 9.00 or later includes a module for discovering NNMi. No additional deployment is necessary.

This section includes the following topics:

- "Use Cases" on page 370
- "Supported Versions" on page 370

Use Cases

This document is based on the following use cases:

- **Use Case 1:** A UCMDB user wants to view the Layer 2 network topology supporting servers and applications. The requirement is to use NNMi as the authoritative source for that information with access through the Universal CMDB application.
- **Use Case 2:** An NNMi operator wants to view the impact of a network access switch infrastructure failure where the impact data is available in UCMDB. The NNMi operator selects an incident or a node in NNMi and then enters a request for impacted CIs.

Supported Versions

Out of the box, the following software versions are supported:

- Data Probe version 9.00 (with Content Pack 4.00) or later
- HP NNMi version 8.11 or later

The following versions are supported after certain updates have been made (as per technical article KM629927 on the HP Support Web site at <http://support.openview.hp.com>):

- Discovery and Dependency Mapping (DDM) Probe, versions 8.0 through 8.04.
- HP NNMi version 8.10.

To use these versions, you must first update the **nnm_sdk.jar** file as directed by HP Software Support.

NNMi - UCMDB Integration Architecture



Tasks

Set Up HP NNMi–HP UCMDB Integration

The following steps describe how to configure NNMi to communicate with UCMDB:

- ▶ "Configure the Connection between NNMi and UCMDB" on page 372
- ▶ "Customize the Integration" on page 373

Configure the Connection between NNMi and UCMDB

On the NNMi management server, do the following:

- 1** In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
- 2** Select the **Enable Integration** check box to activate the remaining fields on the form.
- 3** Enter the information for connecting to the NNMi management server. For information about these fields, see "NNMi Management Server Connection" on page 386.
- 4** Enter the information for connecting to the UCMDB server. For information about these fields, see "UCMDB Server Connection" on page 387.
- 5** Click **Submit** at the bottom of the form.

A new window displays a status message. If the message indicates a problem with connecting to the UCMDB server, re-open the **HP NNMi–HP UCMDB Integration Configuration** form (or press **ALT+LEFT ARROW** in the message window), and then adjust the values for connecting to the UCMDB server as suggested by the text of the error message.

Customize the Integration

On the NNMi management server, do the following:

- 1 In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
- 2 Enter values for the following fields:
 - HP UCMDB Correlation Rule Prefix
 - HP UCMDB Impact Severity Level (1–9)

For details on these fields, see "Integration Behavior" on page 388.

- 3 Click **Submit** at the bottom of the form.

Run HP NNMi–UCMDB Integration

This task includes the steps to run the NNMi/Universal CMDB integration jobs.

Important: To avoid conflict, do not run the UCMDB Layer 2 discovery jobs when running the NNMi Layer 2 integration discovery.

This task includes the following steps:

- "Prerequisites" on page 374
- "Set Up the NNMi Protocol" on page 376
- "Activate the Discovery Jobs" on page 377
- "Check Messages for Successful Job Execution" on page 379
- "Topology Map and Validation of Results" on page 380

1 Prerequisites

- ▶ Ensure that the Data Probe has been installed, as detailed in the *HP Universal CMDB Data Flow Management Guide*.
- ▶ NNMi integration jobs are triggered against the IP CI of the NNMi server. This IP CI must be present in UCMDB. This IP CI may be discovered in one of the following ways:
 - ▶ "Discover the IP CI of the NNMi Server in UCMDB" on page 374
 - ▶ "Manually Add the IP CI of the NNMi Server" on page 375

After the IP CI has been discovered, perform the step "Verify CI Discovery" on page 376.

Note: If you installed HP Business Service Management or HP Operations Manager *i*, you may have installed a bundled UCMDB that uses a Foundation license. If your UCMDB installation has a Foundation license deployed, it is not possible to discover the IP CI automatically. Therefore, you should create this CI manually in the CMDB, as described in "Manually Add the IP CI of the NNMi Server."

Discover the IP CI of the NNMi Server in UCMDB

To add the IP of the NNMi server to the Data Flow Probe range:

- 1 Navigate to **Data Flow Management > Data Flow Probe Setup**.
- 2 Select the Probe that is to be used for the NNMi integration, and add the IP address of the NNMi server to its range.

To discover the IP CI of the NNMi server:

- 1 Navigate to **Data Flow Management > Discovery Control Panel**.
- 2 In the **Network - Basic** discovery module, select the job **Range IPs by ICMP** and click the **Properties** tab.
- 3 In the **Range** parameter line, select the **Override** check box and add the IP address of the NNMi server.

- 4 Click **OK** to save the job, and then activate it to discover the IP CI of the NNMi server.

Manually Add the IP CI of the NNMi Server

Note: When you installed HP Business Service Management or HP Operations Manager *i*, you may have installed a bundled UCMDB that uses a Foundation license. If your UCMDB installation has a Foundation license deployed, use the steps in this section to manually add an IP CI. If any other license (Basic or Advanced) is deployed on the UCMDB server, use the "Discover the IP CI of the NNMi Server in UCMDB" procedure.

To manually add the IP CI of the NNMi server

- 1 Verify that the Data Probe is correctly installed and connected to the UCMDB server.
- 2 Add the IP of the NNMi server to the Data Probe range:
 - a Navigate to **Data Flow Management > Data Flow Probe Setup**.
 - b Select the Probe that is to be used for the NNMi integration, and add the IP address of the NNMi server to its range.
- 3 Insert the IP CI of the NNMi server in the CMDB:
 - a Navigate to **Modeling > IT Universe Manager**.
 - b In the **View** drop-down menu of the CI Selector's View Browser, select **Network Topology**.
 - c Click the **New CI** button.
 - d In the **New CI** dialog box, select the **IP** CIT from the tree and enter the following values:

Field	Description
IP Address	The IP address of the NNMi server.

Field	Description
IP Domain Name	The UCMDB domain name (for example, DefaultDomain).
IP Probe Name	The name of the Data Probe (for example, DefaultProbe).

- e Click **Save** to save the IP CI.

Verify CI Discovery

Note: Verification of CI discovery is relevant only when the IP CI of the NNMi server is discovered (as described in "Discover the IP CI of the NNMi Server in UCMDB" on page 374), not when it is added manually.

In HP Universal CMDB, verify that the following CIs have been discovered before running the NNMi discovery:


- ▶ The **IP** CI of the NNMi server (through the ICMP jobs)
- ▶ The **Host** CI of the NNMi server (through the Host Connection jobs)
- ▶ The **Process** CI of the NNMi Host (through the Host Resource jobs)

To activate a job, select it and click the **Activate** button. For an explanation of a discovery job, see “Jobs” in the *HP Universal CMDB Data Flow Management Guide*.

2 Set Up the NNMi Protocol

In this step, you configure an NNMi protocol entry. This enables the UCMDB server to access information on the NNMi server.

- a Click **Data Flow Management > Data Flow Probe Setup** to open the Data Flow Probe Setup window.
- b If no Probe exists, select **Domains and Probes** and click the **Add Domain or Probe** button to open the Add New Domain dialog box.
- c Enter a name for the new domain and click **OK**.

- d** In the Domains and Probes tree, navigate to the Probe for which you want to set up the NNMi protocol, and click **Credentials**.
- e** Select **NNM Protocol** and click .
- f** Set the protocol attributes, as detailed in "NNMi Protocol Connection Parameters" on page 389, and click **OK**.

For further information about setting up protocols, see "Domain Credential References" in *HP Universal CMDB Data Flow Management Guide*.

1 Activate the Discovery Jobs

The NNMi jobs are included in UCMDB's **Integration – NNM Layer 2** module.

► Layer 2 by NNM job

This job connects to the NNMi Web service and retrieves NNMi discovered nodes, IPs, networks, interfaces, and Layer 2 connection information to create a Layer 2 topology in UCMDB. The job is activated against the **IP** CI of the NNMi server (discovered in the "Verify CI Discovery" step above).

Note: Due to the large volume of data discovered by this DFM job, it may take a while for the Probe to send the data back to the server. If there are more than 20,000 CIs, the Probe returns data in chunks of 20,000 objects at a time.

To activate the Layer 2 by NNM job:

- a** Navigate to **Data Flow Management > Discovery Control Panel**.
- b** In the **Integration - NNM Layer2** discovery module, select the job **Layer2 by NNM** and click the **Properties** tab.
- c** Right-click the job name and select **Activate**.

Note: Steps d through f are necessary only if the NNMi server's IP address CI was added manually, as described in "Manually Add the IP CI of the NNMi Server" on page 375. If the server's IP address was discovered by DFM (as described in "Discover the IP CI of the NNMi Server in UCMDB" on page 374), skip steps d through f.

- d** When the job has been activated, click the **Add CI** button.
- e** Search for the IP CI of the NNMi server and click **Add** to add the IP CI of the NNMi server to the triggered CIs section.
- f** Click **Close** to close the **Choose CIs to Add** dialog box. This causes the job to be activated against the selected IP CI of the NNMi server.

► **Update Ids in NNM job**

This job updates the nodes in the NNMi topology with the UCMDB IDs of the corresponding nodes in UCMDB. This job retrieves the UCMDB IDs of the NNMi hosts from the UCMDB server using the UCMDB Web Services API. The job then updates the **UCMDB_ID** custom attribute on the corresponding node object on the NNMi server using the NNMi Web service. Because the NNMi Web service enables updating of only one node at a time, this process might take a while, depending on the number of nodes involved. Check **probeMgr-adaptersDebug.log** for the update status.

To activate the Update Ids in NNM job:

- a** Navigate to **Data Flow Management > Discovery Control Panel**.
- b** In the **Integration - NNM Layer2** discovery module, select the job **Update Ids in NNM**.
- c** Right-click the job name and select **Activate**.

Note: Steps d through f are necessary only if the NNMi server's IP address CI was added manually, as described in "Manually Add the IP CI of the NNMi Server" on page 375. If the server's IP address was discovered by DFM (as described in "Discover the IP CI of the NNMi Server in UCMDB" on page 374), skip steps d through f.

- d** When the job has been activated, click the **Add CI** button.
- e** Search for the IP CI of the NNMi server and click **Add** to add the IP CI of the NNMi server to the triggered CIs section.
- f** Click **Close** to close the **Choose CIs to Add** dialog box. This causes the job to be activated against the selected IP CI of the NNMi server.

1 Check Messages for Successful Job Execution

You can monitor the **WrapperProbeGw.log** file for job invocation, execution (and possible error) messages. For further debugging information, check the **probeMgr-adaptersDebug.log** file, located in **C:\hp\UCMDB\DataFlowProbe\root\logs**.

The following example shows typical successful job execution messages for the **Layer 2 by NNM** job:

```
- The Job 'NNM Layer 2' started invocation (on 1 destinations)
- Starting NNM_Integration_Utills:mainFunction
- Server: it2tst10.cnd.hp.com, Port: 80, Username: system, MaxPerCall: 2500,
MaxObjects: 50000
- Service URL:
http://it2tst10.cnd.hp.com:80/IPv4AddressBeanService/IPv4AddressBean
- Service URL: http://it2tst10.cnd.hp.com:80/NodeBeanService/NodeBean
- Service URL: http://it2tst10.cnd.hp.com:80/IPv4SubnetBeanService/IPv4SubnetBean
- Service URL: http://it2tst10.cnd.hp.com:80/InterfaceBeanService/InterfaceBean
- Service URL:
http://it2tst10.cnd.hp.com:80/L2ConnectionBeanService/L2ConnectionBean
- OSHVector contains 45426 objects.
- The probe is now going to send back 45426 objects.
- This transfer may take more time than normal due to the large amount of data being
sent to the server.
```

The following example shows typical successful job execution messages for the **Update Ids in NNM** job:

```
- The Job 'NNM Update IDs' started invocation (on 1 destinations)
- UCMDB Server: ucmdb75.fkam.cup.hp.com, UCMDB Port: 8080, UCMDB Username:
admin, UCMDB Protocol: http, UCMDB Context: /axis2/services/UcmdbService
- NNM Server: it2tst10.cnd.hp.com, NNM Port: 80, NNM Username: system
- Getting ready to update Custom Attribute UCMDB_ID on 8161 NNM nodes in NNM
- This process may take a while since the UCMDB_ID custom attribute in NNM can only
be updated one node at a time. Check probeMgr-adaptersDebug.log for status update.
```

2 Topology Map and Validation of Results

Verify that data was discovered using the NNMi integration jobs.

a For a **Layer 2 by NNM** job:

- ▶ In UCMDB, navigate to **Admin > Modeling > IT Universe Manager**.
- ▶ In the **View** drop-down menu of the CI Selector's View Browser, select **Layer 2**. This view displays the CIs and relationships discovered by the integration job.



b For an **Update Ids in an NNM** job:

- ▶ In NNMi, open an NNMi node that was discovered in UCMDB.

- ▶ On the **Custom Attributes** tab, look for the **UCMDB_ID** custom attribute, which should contain the UCMDB ID of the corresponding host in UCMDB.

Use the HP NNMi–HP UCMDB Integration

When you have set up the HP NNMi–HP UCMDB integration, the following URL actions are added to the NNMi console:

- ▶ The **Find UCMDB Impacted CIs** action, which is described in "View Impacted CIs" on page 381.
- ▶ The **Open CI in UCMDB** action, which is described in "View the UCMDB CI" on page 383.

For information about using the integration from the UCMDB user interface, see "Run HP NNMi–UCMDB Integration" on page 373.

View Impacted CIs

Testing for impacted configuration items in UCMDB involves firing a test event of the designated severity and then evaluating the specified correlation rules to determine if the event impacts any other configuration items.

For example:

- ▶ Correlation rule 1 might specify the following impacts:
 - ▶ If Router A experiences a management event of severity 8, Router B and Router C are impacted.
 - ▶ If Router A experiences a management event of severity 9, Router B, Router C, and Router D are impacted.
- ▶ Correlation rule 2 might specify the following impact:
 - ▶ If Router A experiences a management event of any severity, Service E is impacted.

The results of impact analysis on Router A are as follows:

- For a management event of severity 1–7, Service E would be impacted.
- For a management event of severity 8, Router B, Router C, and Service E would be impacted.
- For a management event of severity 8, Router B, Router C, Router D, and Service E would be impacted.

For more information about correlation rules, see “Correlation Manager” in *Model Management*.

For the HP NNMi–HP UCMDB integration, the parameters described in “Integration Behavior” on page 388 specify the severity of the test event and the group of UCMDB correlation rules to evaluate.

The **Find UCMDB Impacted CIs** action displays a list of the UCMDB configuration items that would be impacted for the selected node or interface according to the values of the HP UCMDB Correlation Rule Prefix and HP UCMDB Impact Severity Level (1–9) parameters.

The **Find UCMDB Impacted CIs** action is available from the following NNMi console locations:

- Any node inventory view
- Any interface inventory view
- Any map view (with a node or interface selected)
- Any incident browser

Note: The **Find UCMDB Impacted CIs** action is available for all nodes and interfaces in the NNMi topology, regardless of whether these objects are modeled in the UCMDB database.

View the UCMDB CI

To launch the UCMDB information for a specific CI, select that CI in the HP UCMDB Impacted CIs window (the results of the **Find UCMDB Impacted CIs** action), and then click **Actions > Open CI in UCMDB**.



Note: Since UCMDB is not supported on FireFox, this cross launch works only if NNMi is running in Internet Explorer.

Change the HP NNMi–HP UCMDB Integration Configuration

To update the HP NNMi–HP UCMDB Integration Configuration, perform the following steps:

- 1** In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
- 2** Modify the values as appropriate. For information about the fields on this form, see "HP NNMi–HP UCMDB Integration Configuration Form Reference" on page 385.
- 3** Verify that the **Enable Integration** check box at the top of the form is selected, and then click **Submit** at the bottom of the form.

Note: The changes take effect immediately. You do not need to restart **ovjboss**.

Disable HP NNMi–HP UCMDB Integration Configuration

To disable the HP NNMi–HP UCMDB Integration Configuration, perform the following steps:

- 4** In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
- 5** Clear the **Enable Integration** check box at the top of the form, and then click **Submit** at the bottom of the form. The integration URL actions are no longer available.

Note: The changes take effect immediately. You do not need to restart **ovjboss**.

Perform Impact Analysis

You run impact analysis on a node in NNMi. Use the Universal CMDB Web Services API to call the NNMi correlations in the **NNM_Integration.zip** package:

- NNM_Application_impacts_Application
- NNM_Host_impacts_Application
- NNM_Switch_Router_impacts_Host

For details on running impact analysis, refer to the NNMi documentation. For details on the Universal CMDB Web Services API, see “The HP Universal CMDB Web Service API” in the *HP Universal CMDB Developer Reference Guide*. For details on correlation, see “Correlation Manager” in the *Modeling Guide*.

Reference

HP NNMi–HP UCMDB Integration Configuration Form Reference

The HP NNMi–HP UCMDB Integration Configuration form contains the parameters for configuring communications between NNMi and UCMDB. This form is available from the Integration Module Configuration workspace.

Note: Only NNMi users with the Administrator role can access the HP NNMi–HP UCMDB Integration Configuration form.

The HP NNMi–HP UCMDB Integration Configuration form collects information for the following general areas:

- "NNMi Management Server Connection" on page 386
- "UCMDB Server Connection" on page 387

- ▶ "Integration Behavior" on page 388

To apply changes to the integration configuration, update the values on the **HP NNMi-HP UCMDB Integration Configuration** form, and then click **Submit**.

This section also includes the following topics:

- ▶ "NNMi Management Server Connection" on page 386
- ▶ "UCMDB Server Connection" on page 387
- ▶ "Integration Behavior" on page 388

NNMi Management Server Connection

The following table lists the parameters for connecting to the NNMi management server. This is the same information that you use to open the NNMi console. You can determine many of these values by examining the URL that invokes an NNMi console session. Coordinate with the NNMi administrator to determine the appropriate values for this section of the configuration form.

The default NNMi configuration uses http for connecting to the NNMi console. For information about configuring this connection to use https, see the chapter about enabling https for NNMi in the *HP Network Node Manager i-series Software Deployment Guide*.

Field	Description
HP NNMi SSL Enabled	<p>The connection protocol specification.</p> <ul style="list-style-type: none"> ▶ If the NNMi console is configured to use https, select the NNMi SSL Enabled check box. ▶ If the NNMi console is configured to use http, clear the NNMi SSL Enabled check box. This is the default configuration.
HP NNMi Host	<p>The fully-qualified domain name of the NNMi management server. This field is pre-filled with host name that was used to access the NNMi console. Verify that this value is the name that is returned by the nnmofficialfqdn.ovpl -t command run on the NNMi management server.</p>

Field	Description
HP NNMi Port	<p>The port for connecting to the NNMi console. This field is pre-filled with the port that the jboss application server uses for communicating with the NNMi console, as specified in the following file:</p> <ul style="list-style-type: none"> ▶ <i>Windows:</i> %NnmDataDir%\shared\nnm\conf\nnm.ports.properties ▶ <i>UNIX:</i> \$NnmDataDir/shared/nnm/conf/nnm.ports.properties <p>For non-SSL connections, use the value of jboss.http.port, which is 80 or 8004 by default (depending on the presence of another Web server when NNMi was installed).</p> <p>For SSL connections, use the value of jboss.https.port, which is 443 by default.</p>
HP NNMi User	The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role.
HP NNMi Password	The password for the specified NNMi user.

UCMDB Server Connection

The following table lists the parameters for connecting to the Web services on the UCMDB server. Coordinate with the UCMDB administrator to determine the appropriate values for this section of the configuration.

Field	Description
HP UCMDB SSL Enabled	<p>The connection protocol specification for connecting to the UCMDB Web services.</p> <ul style="list-style-type: none"> ▶ If the UCMDB Web services are configured to use https, select the HP UCMDB SSL Enabled check box. ▶ If the UCMDB Web services are configured to use http, clear the HP UCMDB SSL Enabled check box. This is the default configuration.
HP UCMDB Host	The fully-qualified domain name of the UCMDB server.

Field	Description
HP UCMDB Port	The port for connecting to the UCMDB Web services. If you are using the default UCMDB configuration, use port 8080 (for non-SSL connections to UCMDB).
HP UCMDB User	A valid UCMDB user account name with the UCMDB Administrator role.
HP UCMDB Password	The password for the specified UCMDB user.

Integration Behavior

The following table lists the parameters that describe the integration behavior. Coordinate with the UCMDB administrator to determine the appropriate values for this section of the configuration.

Field	Description
HP UCMDB Correlation Rule Prefix	The prefix of the UCMDB correlation rules that the Find UCMDB Impacted CIs action runs to calculate impact. The default prefix of NNM_ corresponds to the default UCMDB impact correlation rules in the integration package provided by UCMDB (the NNM_Integration.zip file).
HP UCMDB Impact Severity Level (1–9)	The severity level at which to apply the UCMDB impact correlation rules. HP recommends using the highest severity, 9, to include all rules that start with the specified HP UCMDB Correlation Rule Prefix in the calculation of possible impact.

■ NNMi Protocol Connection Parameters

The following table lists the connection parameters from DFM to NNMi.

Field	Description
Connection Timeout	The timeout (in milliseconds) after which the Data Probe stops trying to connect to the NNMi server.
NNM Password	The password for the specified NNMi Web service (for example, Openview).
NNM Username	The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role.
NNM Webservice Port	<p>The port for connecting to the NNMi console. This field is pre-filled with the port that the jboss application server uses for communicating with the NNMi console, as specified in the following file:</p> <ul style="list-style-type: none"> ▶ Windows: %NnmDataDir%\shared\nnm\conf\nnm.ports.properties ▶ UNIX: \$NnmDataDir/shared/nnm/conf/nnm.ports.properties <p>For non-SSL connections, use the value of <code>jboss.http.port</code>, which is 80 or 8004 by default (depending on the presence of another Web server when NNMi was installed).</p> <p>For SSL connections, use the value of <code>jboss.https.port</code>, which is 443 by default.</p>
NNM Webservice Protocol	The protocol for the NNMi Web service (the default is <code>http</code>).
UCMDB Password	The password for the UCMDB Web service (the default is <code>admin</code>).
UCMDB Username	A valid UCMDB Web service account name with the UCMDB Administrator role (the default is <code>admin</code>).

Field	Description
UCMDB Webservice Port	The port for connecting to the UCMDB Web service. If you are using the default UCMDB configuration, use port 8080 (for non-SSL connections to UCMDB).
UCMDB Webservice Protocol	The connection protocol specification for connecting to the UCMDB Web services. <ul style="list-style-type: none"> ▶ If the UCMDB Web services are configured to use https, select the HP UCMDB SSL Enabled check box. ▶ If the UCMDB Web services are configured to use http, clear the HP UCMDB SSL Enabled check box. This is the default configuration.

Troubleshooting and Limitations

- ▶ **Problem.** The NNMi Web service responds with a **cannot interrogate model** message.

Solution. This message usually indicates that the Web services request made to the NNMi server is incorrect or too complex to process. Check the NNMi jbossServer.log file for details.

- ▶ **Problem.** If an excessive number of nodes are to be updated with the same UCMDB ID, it may take a while for the update pattern to complete.

Solution. The volume of data retrieved from the NNMi server might be large. The recommended memory requirements for the Data Probe process is 1024 MB. Since the NNMi Web service enables updating the individual nodes one at a time, the time to update the nodes may take a while.

- ▶ **Problem.** You have verified the values in the **HP NNMi–HP UCMDB Integration Configuration** form, but the status message still indicates a problem with connecting to the UCMDB server.

Solution.

- a Clear the Web browser cache.
- b Clear all saved form or password data from the Web browser.

- c Close the Web browser window completely, and then re-open it.
- d Re-enter the values in the **HP NNMi–HP UCMDB Integration Configuration** form.

- **Problem.** The **Layer 2 by NNM** job finishes with the following warning:
Failed to get any Layer 2 links from NNM.

Solution. Refer to technical article KM629927 on the HP support Web site at <http://support.openview.hp.com>.

- **Problem.** Either of the NNMi integration jobs fails with the following error in the DFM log files: com.hp.ov.nms.sdk.node.NmsNodeFault: Cannot interrogate model.

Solution. This error typically means that the NNMi server failed to process the Web services call. Check the following two logs on the NNMi server for exceptions when the integration was activated:

- jbossServer.log
- sdk.0.0.log

- **Problem.** Either of the NNMi integration jobs fail with the following error: Could not find Discovery Probe 'DefaultProbe'. Task for TriggerCI will not be created.

Solution.

- a Right-click the job and select **Go To Pattern**.
- b Click the **Pattern Management** tab.
- c Select the **Override default Probe selection** check box, and enter the name of the Probe used for the NNMi integration in the **Probe** field.
- d Click **Save** to save the pattern, then reactivate the job against the **IP CI** of the NNMi server.

33

Storage Essentials (SE) Integration with HP Universal CMDB

This chapter includes:

Concepts

- ▶ SE Integration – Overview on page 393

Tasks

- ▶ Discover the SE Oracle Database on page 395

Reference

- ▶ Storage Essentials Integration Packages on page 398
- ▶ Discovered CITs on page 398
- ▶ Views on page 403
- ▶ Correlation Rules on page 407
- ▶ Reports on page 409

Concepts

SE Integration – Overview

Integration involves synchronizing devices, topology, and the hierarchy of a customer storage infrastructure in the Universal CMDB database (CMDB). This enables Change Management and Impact Analysis across all business services mapped in UCMDB from a storage point of view.

You integrate SE with UCMDB using Data Flow Management.

When you activate the **Integration – Storage Essentials** module, DFM retrieves data from the SE Oracle database and saves CIs to the Universal CMDB database. Users can then view SE storage infrastructure in UCMDB.

The data includes information on storage arrays, fiber channel switches, hosts (servers), storage fabrics, logical volumes, host bus adapters, storage controllers, and fiber channel ports. Integration also synchronizes physical relationships between the hardware, and logical relationships between logical volumes, storage zones, storage fabrics, and hardware devices.

Note: DFM version 9.00 or later includes a module for discovering SE. No additional deployment is necessary.

Supported Versions

The integration procedure supports DFM version 9.00 or later and SE version 6.x.

SE Installation Requirements

The minimum VM installation requirements for SE integration are:

- 4 GB memory
- 50 GB hard drive space

Tasks

Discover the SE Oracle Database

This task includes the steps to run the SE/UCMDB integration jobs.

This task includes the following steps:

- "Prerequisites" on page 395
- "Network and Protocols" on page 396
- "Activate the Discovery Job" on page 396
- "Topology Map" on page 397

1 Prerequisites

In DFM, in the Discovery Control Panel window, verify that the following CIs have been discovered before running the SE discovery:

- **Network Discovery > Basic > Class C IPs by ICMP or Range IPs by ICMP:** discovers the IP address of the Oracle database server
- **Database > Oracle > Database TCP Ports:** discovers TCP ports on the IP address discovered previously
- **Database > Oracle > Oracle Database Connection by SQL:** discovers Oracle server instances
- **Discovery Based Product Integrations > Storage Essentials > SE Integration by SQL:** discovers storage infrastructure

For details on activating a job, see "Discovery Modules Pane" in *HP Universal CMDB Data Flow Management Guide*. For an explanation of a discovery job, see "Discovery Jobs" in *HP Universal CMDB Data Flow Management Guide*.

Note:

- ▶ For the **Oracle Connection by SQL** job, it is recommended to use the **REPORT_USER** Oracle user name, since this user has privileges necessary to run SQL queries on the APPIQ_SYSTEM tables.
 - ▶ This DFM job queries Oracle Materialized Views, and the views may be in the process of being refreshed when the DFM job is executed. This could result in an error message identifying the problem and a request to run the job later.
-

2 Network and Protocols

SE uses the **SQL protocol**. For details, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*.

3 Activate the Discovery Job

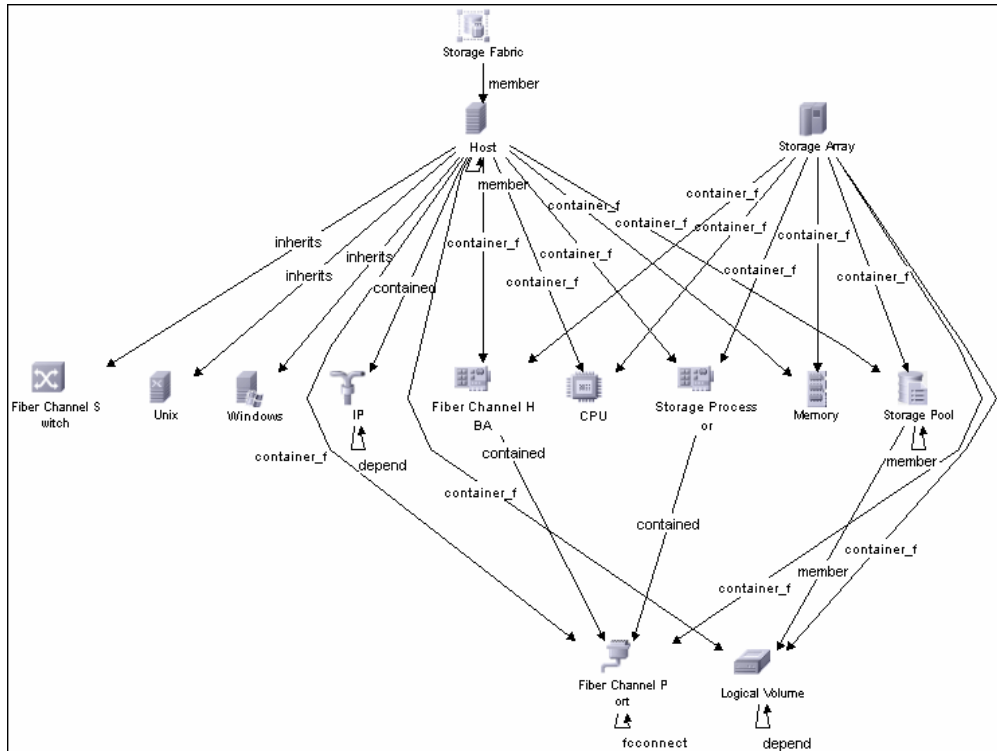
The **SE Integration by SQL** job is included in the **Integration – Storage Essentials** module.

The **SE Integration by SQL** job runs queries against Oracle materialized views, installed and maintained by Storage Essentials in the Oracle database. The job uses a database CI as the trigger.

For details on activating a job, see "Discovery Modules Pane" in *HP Universal CMDB Data Flow Management Guide*.

4 Topology Map

The following diagram illustrates the CITs discovered by the SE Integration by SQL job:



Reference

Storage Essentials Integration Packages

The integration includes two UCMDB packages:

- ▶ **SE_Discovery.zip.** Contains the trigger TQL for SE discovery, discovery script, adapter, and job. The discovery adapter has no parameters and requires no configuration.
- ▶ **Storage_Basic.zip.** Contains the new CI Type definitions, views, reports, and correlation rules. This package is common to all Storage Management integration solutions.

Tip: You can include the SE job in the DFM schedule. For details, see "Discovery Scheduler Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.

Discovered CITs

The following CITs represent SE storage entities in UCMDB:

- ▶ **Fiber Channel Connect.** This CIT represents a fiber channel connection between fiber channel ports.
- ▶ **Fiber Channel HBA.** This CIT has change monitoring enabled on parameters such as state, status, version, firmware version, driver version, WWN, and serial number. A Fiber Channel HBA inherits from the Host Resource CIT.
- ▶ **Fiber Channel Port.** This CIT has change monitoring enabled on parameters such as state, status, WWN, and trunked state. Since a Fiber Channel Port is a physical port on a switch, it inherits from the Physical Port CIT under the Network Resource CIT.

- ▶ **Fiber Channel Switch.** A switch falls under the Host CIT since SE maintains an IP address for each switch. Parameters such as status, state, total/free/available ports, and version are change monitored.

This package retrieves Fiber Channel Switch details from the **mvc_switchsummaryvw** and **mvc_switchconfigvw** views. The discovery retrieves detailed information about Fiber Channel Ports on each switch from the **mvc_portsummaryvw** view.

A switch inherits from a Host CIT in UCMDB. Since DFM uses the IP address of a host as part of its primary key, this DFM job attempts to use an IP address from SE for this purpose. If an IP address is not available, the job attempts to resolve the switch's IP address using a DNS name (also maintained by SE). If neither an IP address nor a DNS name is available, the switch is discarded.

- ▶ **Logical Volume.** This CIT represents volumes on Storage Arrays and hosts with change monitoring on availability, total/free/available space, and storage capabilities.
- ▶ **Storage Array.** This CIT represents a Storage Array with change monitoring on details such as serial number, version, and status. Since a storage array may not have a discoverable IP address, it inherits from the Network Resource CIT.

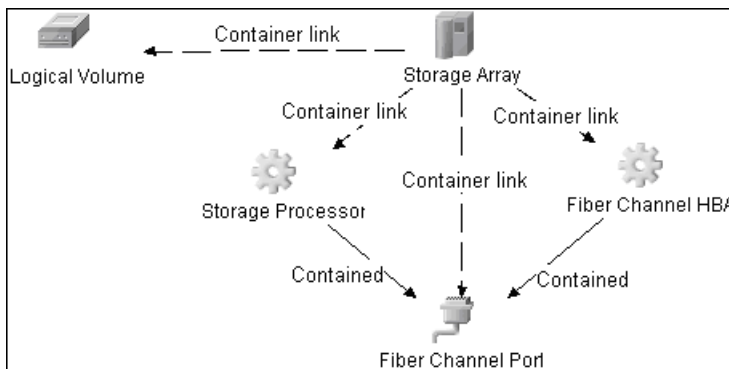
This CIT retrieves Storage Array details from the **mvc_storagesystemssummaryvw** view. DFM retrieves detailed information on Storage Processors and HBAs from the **mvc_storageprocessorssummaryvw** and **mvc_cardsummaryvw** tables respectively.

The SE database may possibly not be able to obtain IP address information on Storage Arrays for a variety of technical and policy related reasons. Since a Storage Array is a host as far as DFM is concerned, DFM assumes that the serial number of a Storage Array is unique and uses this as the primary key. The CI is then manually set as a complete host. If the serial number of a Storage Array is not available, the array is discarded.

Since Fiber Channel Ports may be present on a Storage Array, Storage Processor, or HBA, DFM uses three separate queries to retrieve Fiber Channel Ports for each Storage Array. Detailed information about Fiber Channel Ports on each array are retrieved from the **mvc_portsummaryvw** view. Since this view uses a container ID as the key, DFM queries the view by container ID for each Storage Array, each Storage Processor on a Storage Array, and each HBA on a Storage Array.

DFM retrieves detailed information about Logical Volumes on each Storage Array from the **mvc_storagevolumesummaryvw** view.

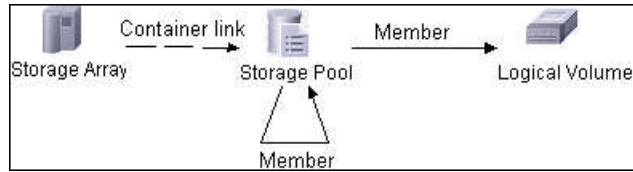
Results from these queries populate a map as shown below:



- ▶ **Storage Fabric.** This CIT inherits from the Network Resource CIT and represents a storage fabric. This CIT has no change monitoring enabled.
- ▶ **Storage Processor.** This CIT represents other storage devices such as SCSI controllers, and inherits from the Host Resource CIT. A Storage Processor CIT monitors change on parameters such as state, status, version, WWN, roles, power management, and serial number.
- ▶ **Storage Pool.**

Storage Pool information is also collected from each Storage Array using the query below.

Results from this query populate a map as shown below:



Host Details

DFM retrieves Host details from the **mvc_hostsummaryvw** view and detailed information on HBAs from the **mvc_cardsummaryvw** view.

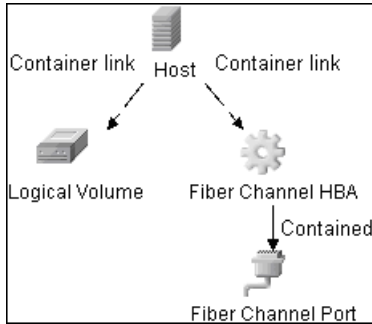
SE maintains information on Operating Systems, Memory, IP address, and DNS name on each host. DFM uses this information to create Host CIs of type UNIX or Windows, and adds Memory CIs for each host as available.

Since UCMDB uses the IP address of a host as part of its primary key, DFM attempts to use the IP address from SE for this purpose. If an IP address is not available, DFM then attempts to resolve the host's IP address using a DNS name. If neither an IP address nor a DNS name is available, DFM ignores the host.

Similar to Storage Arrays, a host may have Fiber Channel Ports directly associated with itself or on HBAs on the host. The DFM job uses three separate queries to retrieve Fiber Channel Ports for each host. The job retrieves detailed information about Fiber Channel Ports on each host from the **mvc_portsummaryvw** view. Since this view uses a ContainerID attribute as the key, the job queries the view by containerID for each host, and each HBA on a host.

Finally, DFM retrieves detailed information about Logical Volumes on each host from the **mvc_hostvolumesummaryvw** and **mvc_hostcapacityvw** views. The **mvc_hostcapacityvw** view maintains capacity information for each volume over multiple instances in time, and the job uses only the latest available information.

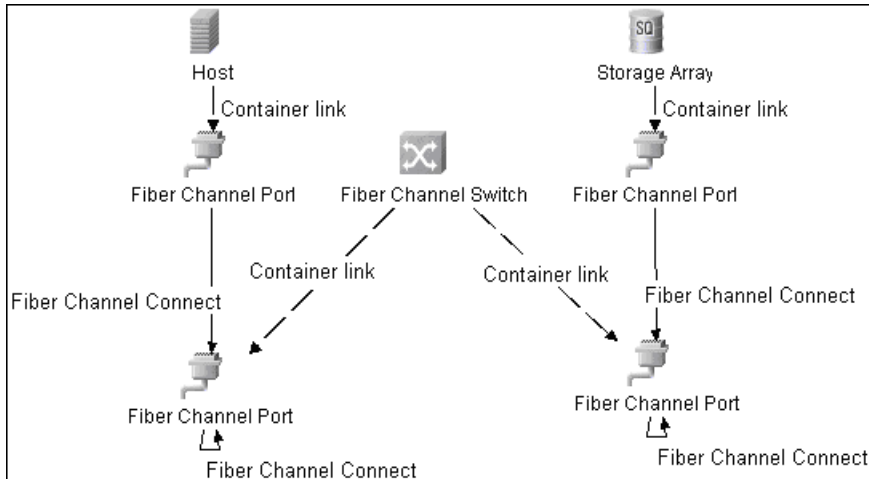
Results from these queries populate a map as shown below:



SAN Topology

SAN Topology consists of the Fiber Channel network topology and includes (fiber channel) connections between Fiber Channel Switches, Hosts, and Storage Arrays. SE maintains a list of WWNs that each Fiber Channel Port connects to, and this package uses this list of WWNs to establish Fiber Channel Connection links.

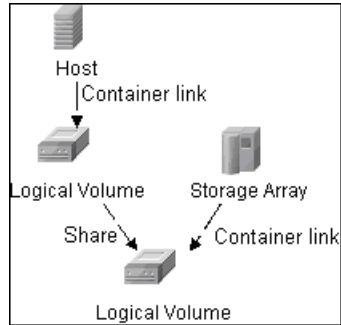
Results from these queries populate a map as shown below:



Storage Topology

Storage topology consists of relationships between Logical Volumes on a host and Logical Volumes on a Storage Array. DFM uses multiple tables to identify this relationship as shown in the query below. This view is a summary of all of the above information.

Results from these queries populate a map as shown below:



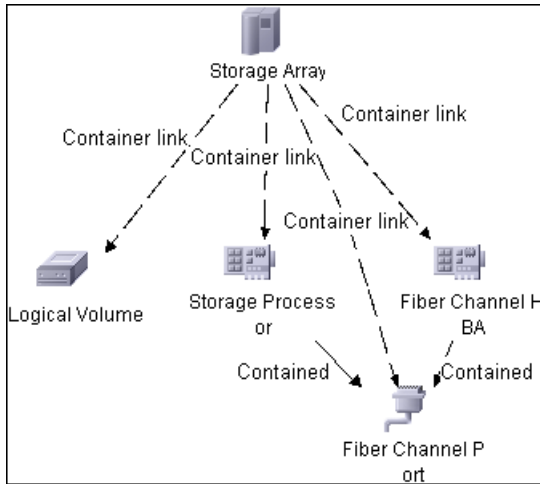
Views

The SE package contains views that display common storage topologies. These are basic views that can be customized to suit the integrated SE applications.

Storage Array Details

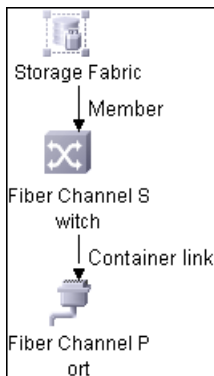
This view shows a Storage Array and its components including Logical Volumes, HBAs, Storage Processors, and Fiber Channel Ports. The view shows each component under its container Storage Array and groups Logical Volumes by CI Type.

Storage Array does not require all components in this view to be functional. Container links stemming from the Storage Array have a cardinality of zero-to-many. The view may show Storage Arrays even when there are no Logical Volumes or Storage Processors.



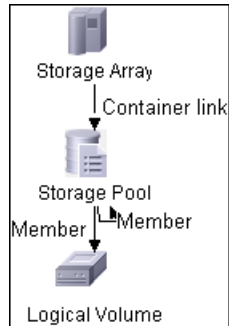
FC Switch Details

This view shows a Fiber Channel Switch and all connected Fiber Channel Ports.



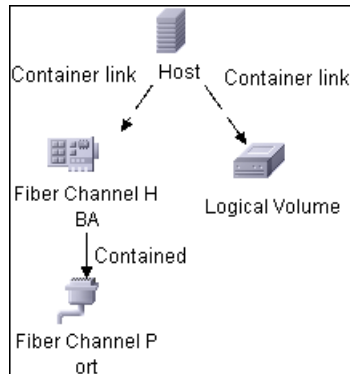
Storage Pool Details

This view shows Storage Pools with associated Storage Arrays and Logical Volumes.



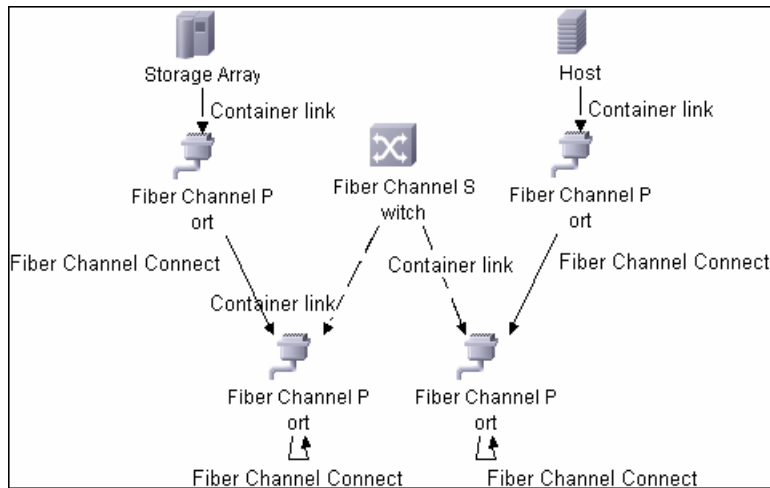
Host Storage Details

This view shows only Hosts that contain a Fiber Channel HBA or a Logical Volume. This keeps the view storage-specific and prevents hosts discovered by other DFM jobs from being included in the view.



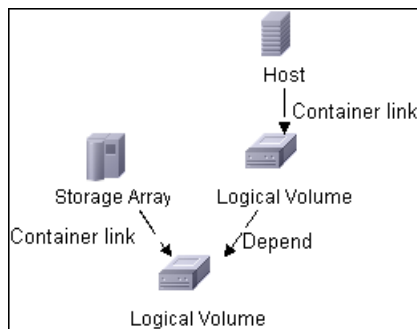
SAN Topology

This view maps physical connections between Storage Arrays, Fiber Channel Switches, and Hosts. The view shows Fiber Channel Ports below their containers. The view groups the Fiber Channel Connect relationship CIT to prevent multiple relationships between the same nodes from appearing in the top layer.



Storage Topology

This view maps logical dependencies between Logical Volumes on Hosts and Logical Volumes on Storage Arrays. There is no folding in this view.



Correlation Rules

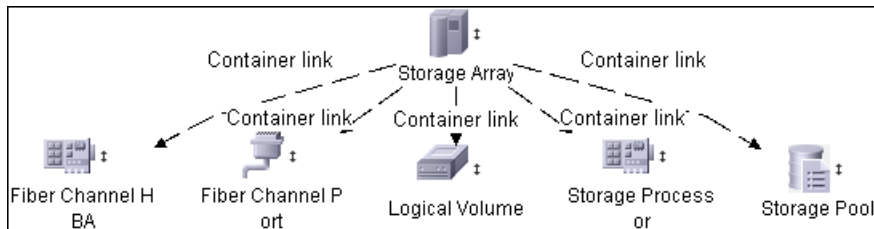
This package contains basic correlation rules to enable impact analysis and root cause analysis in UCMDB. These correlation rules are templates for more complex rules that you can define based on business needs.

All correlation rules fully propagate both Change and Operation events. For details on impact analysis, see "Impact Analysis Manager Page" and "Impact Analysis Manager Overview" in *Modeling Guide*.

Note: Correlation events are not propagated to Fiber Channel Ports for performance reasons.

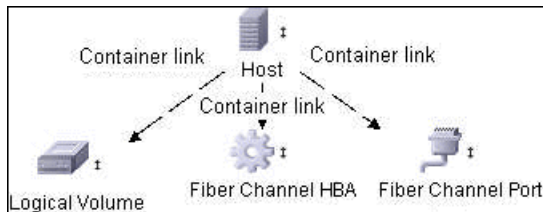
Storage Array Devices to Storage Array

This correlation rule propagates events between Logical Volumes, Storage Processors, Fiber Channel HBAs, and Storage Arrays.



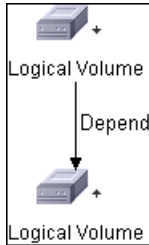
Host Devices to Host

This correlation rule propagates events between Fiber Channel HBAs and Hosts, and Logical Volumes on the Host.



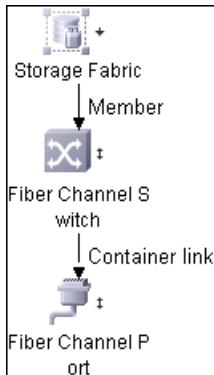
Logical Volume to Logical Volume

This correlation rule propagates events on a Logical Volume contained in a Storage Array to the dependent Logical Volume on the Host.



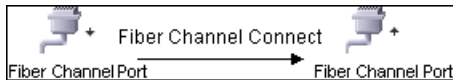
FC Switch Devices to FC Switch

This correlation rule propagates events from a Fiber Channel Port to and from a Switch. The event is also propagated to the associated Storage Fabric.



FC Port to FC Port

This rule propagates events on a Fiber Channel Port to another connected Channel Port.



Example of HBA crashing on a Storage Array:

- ▶ The event propagates from the HBA to the Storage Array and the Logical Volumes on the Array because of the Storage Devices to Storage Array rule.
- ▶ The correlation event on the Logical Volume then propagates to other dependent Logical Volumes through the Logical Volume to Logical Volume rule.
- ▶ Hosts using those dependent Logical volumes see the event next because of the Host Devices to Host rule.
- ▶ Depending on business needs, you define correlation rules to propagate events from these hosts to applications, business services, lines of business, and so on. This enables end-to-end mapping and impact analysis using UCMDB.

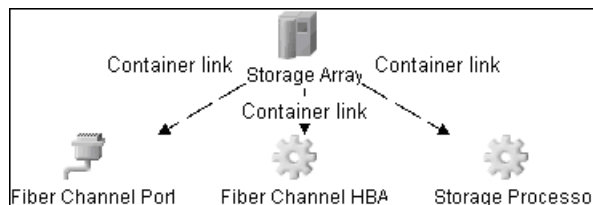
Reports

The SE package contains basic reports that can be customized to suit the integrated SE applications.

In addition to the system reports, Change Monitoring and Asset Data parameters are set on each CIT in this package, to enable Change and Asset Reports in Universal CMDB. For details see "Storage Array Configuration" on page 409, "Host Configuration" on page 410, "Storage Array Dependency" on page 410, and "Host Storage Dependency" on page 410.

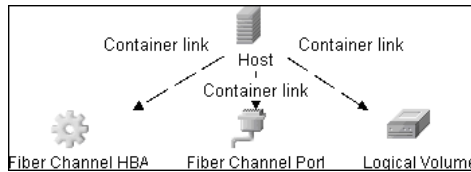
Storage Array Configuration

This report shows detailed information on Storage Arrays and its sub-components including Fiber Channel Ports, Fiber Channel Arrays, and Storage Processors. The report lists Storage Arrays with sub-components as children of the Array.



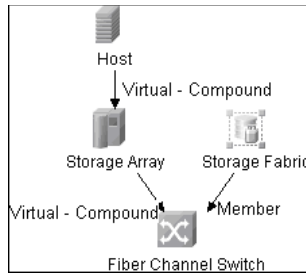
Host Configuration

This report shows detailed information on hosts that contain one or more Fiber Channel HBAs, Fiber Channel Ports, or Logical volumes. The report lists hosts with sub-components as children of the host.



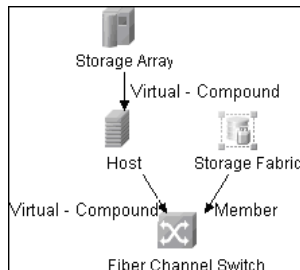
Storage Array Dependency

This report maps dependencies on a Storage Array. The report also displays information on switches connected to it.



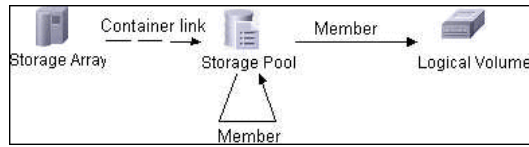
Host Storage Dependency

This report shows detailed information on storage infrastructure dependencies of a Host. The report lists hosts and dependent components.



Storage Pool Configuration

This report shows detailed information on Storage Pool configuration.



34

EMC Control Center (ECC) Integration with HP Universal CMDB

Note: This functionality is available as part of Content Pack 5.00 or later.

This chapter includes:

Concepts

- ECC Integration – Overview on page 414

Tasks

- Discover the ECC Storage Topology on page 415

Reference

- ECC Job SQL Queries on page 422
- Views on page 425
- Correlation Rules on page 429
- Reports on page 431

Concepts

ECC Integration – Overview

Integration between ECC and DFM involves synchronizing devices, topology, and hierarchy of storage infrastructure in the UCMDB database (CMDB). This enables Change Management and Impact Analysis across all business services mapped in UCMDB from a storage point of view.

DFM initiates discovery on the ECC database. Synchronized Configuration Items (CIs) include Storage Arrays, Fiber Channel Switches, Hosts (Servers), Storage Fabrics, Storage Zones, Logical Volumes, Host Bus Adapters, Storage Controllers, and Fiber Channel Ports. The integration also synchronizes physical relationships between hardware, and logical relationships between Logical Volumes and hardware devices, to enable end-to-end mapping of the storage infrastructure.

You integrate ECC with UCMDB using Data Flow Management.

Note: DFM Content Pack version 6.00 or later includes a module for discovering ECC. No additional deployment is necessary.

Tasks

Discover the ECC Storage Topology

This task includes the steps to run the ECC/UCMDB integration job.

This task includes the following steps:

- "Supported Versions" on page 415
- "Prerequisites" on page 415
- "Network and Protocols" on page 418
- "Discovery Workflow" on page 418
- "Discovery Adapter Parameters" on page 419
- "Discovered CIs" on page 419
- "The ECC Integration Package" on page 420
- "Topology Map" on page 421

1 Supported Versions

Target Platform	OS Platform	DFM Protocol	ECC Version	DFM Version
EMC Control Center	All	SQL over JDBC, SSL optional	6.x	DFM Content Pack 6.00 or later

2 Prerequisites

- **Deploy the ECC Integration package**
 - a** If you are connecting to the ECC Oracle database with SSL communication, in DFM populate the SQL protocol or adapter parameters. For details, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*.
 - b** Verify that the IP address of the ECC server is within scope of a Data Flow Probe. For details, see "Add/Edit IP Range Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.

► **Set up SSL communication with the ECC Oracle database**

Caution: Perform this procedure if SSL communication is enabled for the ECC database.

- c** Retrieve the following files from the Oracle server and copy them to **C:\dbSafe** (or another convenient location) on the Data Flow Probe file system:
 - cert.txt or user.crt
 - cwallet.sso
 - ewallet.p12
- d** Populate the **walletLocation** DFM adapter parameter with the absolute location of **cwallet.sso**, for example, **C:\dbSafe\cwallet.sso**. For details on job parameters, see "Discovery Adapter Parameters" on page 419.
- e** Open the **WrapperEnv.conf** file with a text editor. The file is located in **C:\hp\UCMDB\DataFlowProbe\bin**.
- f** Add a new line after line ~82 that looks like:

```
set.ORACLE_SSL_CLASSES=  
%lib%/collectors/probeManager/discoveryResources/db/oracleSSL/oraclepki.jar  
;  
%lib%/collectors/probeManager/discoveryResources/db/oracleSSL/ojpse.jar;  
%lib%/collectors/probeManager/discoveryResources/db/oracleSSL/ojdbc14.jar
```

- g** Append the following code to the end of the line beginning with **set.COMMON_CLASSPATH** (line ~87):

```
;%ORACLE_SSL_CLASSES%
```

- h** Save and close the file.
- i** Restart the Data Flow Probe.

► **Run the DFM jobs**

In DFM, in the Discovery Control Panel window, run one of the following sets of jobs to trigger ECC discovery:

a Set 1:

- **Network Discovery > Basic > Range IPs by ICMP.** Discovers the IP address of the ECC server.
- **Network Discovery > Basic > Host Connection by Shell/WMI/SNMP.** Discovers operating system information on the ECC server.
- **Network Discovery > Host Resources and Applications > Host Resources and Applications by Shell/SNMP/WMI.** Discovers the Oracle database instance used by ECC.
- **Database > Oracle > Oracle Database Connections by SQL.** Discovers Oracle databases using the SQL protocol.

Caution: If you are working with an SSL-enabled database, do not run this job.

b Set 2:

- **Network Discovery > Basic > Range IPs by ICMP.** Discovers the IP address of the ECC server.
- **Database > Oracle > Database TCP ports.**
- **Database > Oracle > Oracle Database Connections by SQL.** Discovers Oracle databases using the SQL protocol.

For details on activating a job, see "Discovery Modules Pane" in *HP Universal CMDB Data Flow Management Guide*. For an explanation of a discovery job, see "Discovery Jobs" in *HP Universal CMDB Data Flow Management Guide*.

3 Network and Protocols

- ▶ In DFM, set up the **SQL protocol**. Populate the parameters with the credentials to the ECC database.

For details, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*.

- ▶ These credentials should have SELECT permissions on the following tables/views:
 - ▶ Fiber channel switches: **STSSYS.STS_SWITCH_LIST**
 - ▶ Fiber channel ports on switches: **STSSYS.STS_SWITCH_PORT**
 - ▶ Storage arrays: **STSSYS.STS_ARRAY_LIST**
 - ▶ Fiber channel ports on arrays: **STSSYS.STS_ARRAY_PORT**
 - ▶ Logical volumes on arrays: **STSSYS.STS_ARRAY_DEVICE**
 - ▶ Hosts/servers: **STSSYS.STS_HOST_LIST**
 - ▶ Fiber channel ports and HBAs on hosts: **STSSYS.STS_HOST_HBA**
 - ▶ Logical volumes on hosts: **STSSYS.STS_HOST_DEVICE**
 - ▶ Logical volume dependencies: **STSSYS.STS_HOST_SHAREDDEVICE**
 - ▶ Port connections: **STSSYS.STS_ARRAY_PORT_CONNECTION**

Note: The ECC database instance has an out-of-the-box user account named **STSVIEW** that includes the necessary privileges. The default password for this account is **sts**.

4 Discovery Workflow

- ▶ Activate the **Integration – EMC Control Center > ECC Integration by SQL** job. This job discovers the storage infrastructure of ECC.

The **ECC Integration by SQL** job runs SQL queries on the ECC Oracle database using JDBC. This Oracle database instance is used as a trigger for the DFM job. For details of the SQL queries, see "ECC Job SQL Queries" on page 422.

Tip: You can include the ECC job in the DFM schedule. For details, see "Discovery Scheduler Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.

5 Discovery Adapter Parameters

Name	Value	Description
useSSL	false	If SSL is required to connect to the ECC database, change to true .
walletLocation	C:\dbSafe\cwallet.sso	The location of the Oracle wallet. This parameter is required for SSL connections and defaults to C:\dbSafe\cwallet.sso .
walletType	SSO	The Oracle wallet type. This parameter is required for SSL connections. The default for the ECC database is SSO .

6 Discovered CIs

- CPU
- Contained (link)
- Container link (link)
- Depend (link)
- Fiber Channel Connect (link)
- Fiber Channel HBA
- Fiber Channel Port
- Fiber Channel Switch
- Host

- IP
- Logical Volume
- Member
- Memory
- Storage Array
- Storage Fabric
- Storage Processor
- Unix
- Windows

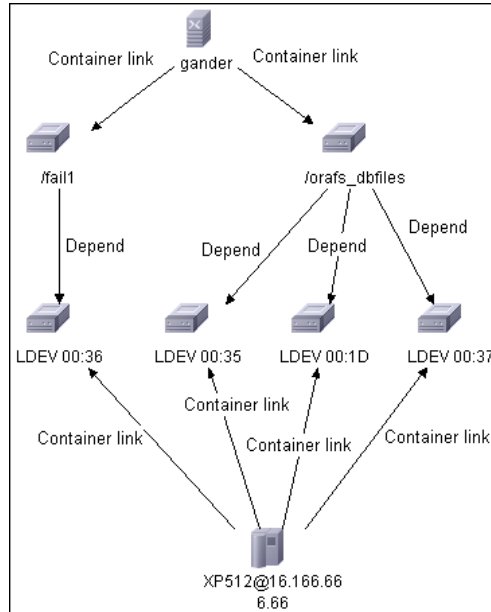
7 The ECC Integration Package

The integration includes the following package:

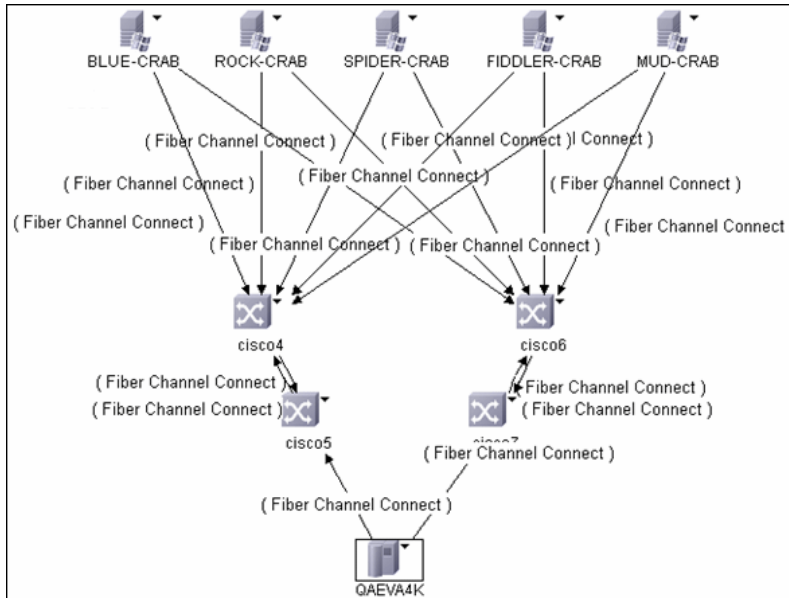
- **ECC_Integration.zip**. Contains the trigger TQL, DFM script, adapter, and job for ECC discovery. The DFM job uses the ECC Oracle database CI as the trigger.

8 Topology Map

The following diagram illustrates the storage topology and shows the relationships between logical volumes on a storage array and those on servers:



The following diagram illustrates the SAN (Storage Area Networks) topology showing fiber channel paths between storage arrays, switches, and servers:



Reference

ECC Job SQL Queries

The following workflow explains how the **ECC Integration by SQL** job discovers the storage topology of ECC. The job:

- 1 Connects to the ECC Oracle database instance using credentials from the SQL protocol. For details, see "Network and Protocols" on page 418.

- 2** Queries for fiber channel switches and ports on each switch and creates **Fiber Channel Switch** CIs:

```
SELECT switch.st_id, switch.st_sn, switch.st_alias, switch.st_model, switch.st_version,
switch.st_vendor, switch.sw_managementurl, switch.sw_domain, switch.sw_portcount,
switch.sw_portcount_free FROM stssys.sts_switch_list switch WHERE
LOWER(switch.sw_principal) = 'true'
```

- 3** Queries for fiber channel adapters and ports on each Fiber Channel Switch and creates **Fiber Channel HBA** and **Fiber Channel Port** CIs:

```
SELECT port.port_id, port.port_number, port.port_type, port.adport_alias,
port.port_wwn, port.port_status, port.conn_port_wwn FROM stssys.sts_switch_port port
WHERE port.st_id = switch.st_id from above query
```

- 4** Queries for storage arrays and creates **Storage Array** CIs:

```
SELECT array.st_id, array.st_sn, array.st_alias, array.st_type, array.st_model,
array.st_vendor, array.st_microcode, array.sy_microcode_patch,
array.sy_microcode_patchdate FROM stssys.sts_array_list array
```

- 5** Queries for Fiber Channel ports, Fiber Channel host bus adapters (HBA), and logical volumes on each storage array, and creates **Fiber Channel Port**, **Fiber Channel Port HBA**, and **Logical Volume** CIs:

```
SELECT port.port_id, port.port_number, port.port_type, port.adport_alias,
port.port_wwn, port.port_status FROM stssys.sts_array_port port WHERE port.st_id =
array.st_id from above query
```

```
SELECT hba.port_id, hba.ad_id, hba.ad_name FROM stssys.sts_array_port hba
WHERE hba.st_id = array.st_id from above query
```

```
SELECT logicalVolume.sd_id, logicalVolume.sd_name, logicalVolume.sd_alias,
logicalVolume.sd_size, logicalVolume.sd_type FROM stssys.sts_array_device
logicalVolume WHERE logicalVolume.st_id = array.st_id from above query
```

- 6** Queries for hosts/servers and creates appropriate **Computer**, **Windows**, or **Unix** CIs. Results of this query are used to create host resource CIs such as **Memory** and **CPU** if this information is available:

```
SELECT host.host_id, host.host_name, host.host_alias, host.host_domain,
host.host_model, host.host_ip, host.host_vendorname, host.host_cpucount,
host.host_installedmemory, host.host_os, host.host_osversion, host.host_oslevel,
host.host_osclass FROM stssys.sts_host_list host
```

- 7** Queries for Fiber Channel ports, Fiber Channel host bus adapters (HBA), and logical volumes on each host/server and creates **Fiber Channel Port**, **Fiber Channel Port HBA**, and **Logical Volume** CIs:

```
SELECT port.port_id, port.port_number, port.adport_alias, port.port_wwn FROM
stssys.sts_host_hba port WHERE port.host_id = host.host_id from above query
```

```
SELECT hba.ad_id, hba.ad_name, hba.fibread_nodewwn, hba.ad_vendor,
hba.ad_revision, hba.ad_model, hba.port_id, hba.ad_driver_rev FROM
stssys.sts_host_hba hba WHERE hba.host_id = host.host_id from above query
```

```
SELECT logicalVolume.hd_id, logicalVolume.hd_name, logicalVolume.hd_type,
logicalVolume.hd_total FROM stssys.sts_host_device logicalVolume WHERE
logicalVolume.hd_id IS NOT NULL AND logicalvolume.arraybod_type = 'Array' AND
logicalVolume.host_id = host.host_id from above query
```

- 8** Queries for logical volume mapping between logical volumes on hosts/servers and logical volumes on storage arrays, and adds **Depend** relationships between hosts/servers and storage arrays:

```
SELECT sd_id FROM stssys.sts_host_shareddevice WHERE hd_id =
logicalvolume.hd_id from above query
```

- 9** Queries for paths between hosts/servers and storage arrays and adds **Fiber Channel Connect** relationships between respective hosts/servers, switches, and storage arrays:

```
SELECT port.port_wwn, port.conn_port_wwn FROM stssys.sts_array_port_connection
port WHERE port.port_wwn IS NOT NULL AND port.conn_port_wwn IS NOT NULL
```

```
SELECT port.port_wwn, port.conn_port_wwn FROM stssys.sts_switch_port port
WHERE port.port_wwn IS NOT NULL AND port.conn_port_wwn IS NOT NULL
```


Views

The **Storage_Basic** package contains views that display common storage topologies. These are basic views that can be customized to suit the integrated ECC applications.

To access the Storage_Basic package: **Administration > Package Manager**. For details, see "Package Manager" in *HP UCMDB Administration Guide*.

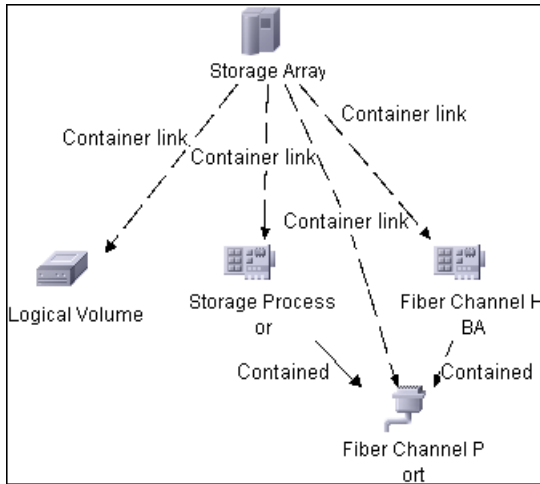
This section includes:

- "Storage Array Details" on page 425
- "FC Switch Details" on page 426
- "Storage Pool Details" on page 427
- "Host Storage Details" on page 427
- "SAN Topology" on page 428
- "Storage Topology" on page 428

Storage Array Details

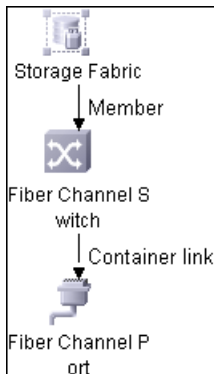
This view shows a Storage Array and its components including Logical Volumes, HBAs, Storage Processors, and Fiber Channel Ports. The view shows each component under its container Storage Array and groups Logical Volumes by CI Type.

Storage Array does not require all components in this view to be functional. Container links stemming from the Storage Array have a cardinality of zero-to-many. The view may show Storage Arrays even when there are no Logical Volumes or Storage Processors.



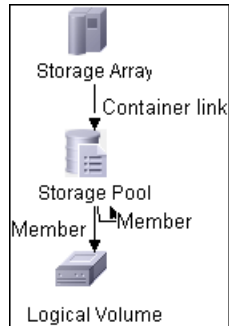
FC Switch Details

This view shows a Fiber Channel Switch and all connected Fiber Channel Ports.



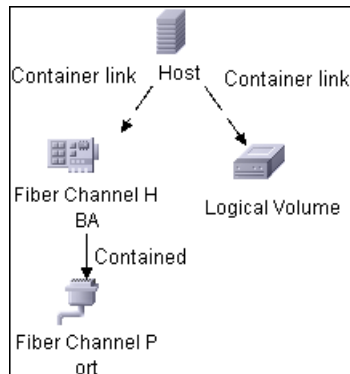
Storage Pool Details

This view shows Storage Pools with associated Storage Arrays and Logical Volumes.



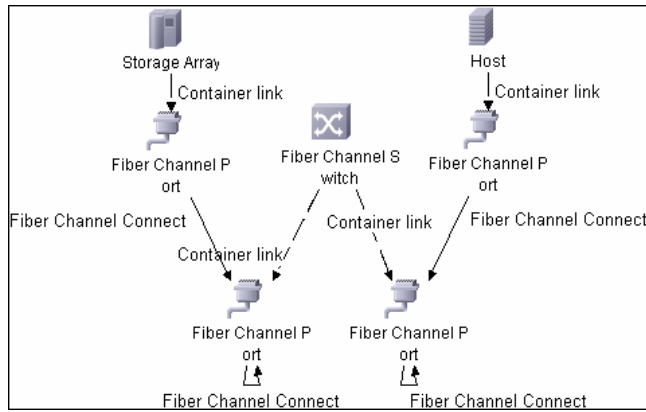
Host Storage Details

This view shows only Hosts that contain a Fiber Channel HBA or a Logical Volume. This keeps the view storage-specific and prevents hosts discovered by other DFM jobs from being included in the view.



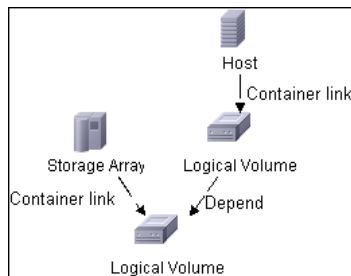
SAN Topology

This view maps physical connections between Storage Arrays, Fiber Channel Switches, and Hosts. The view shows Fiber Channel Ports below their containers. The view groups the Fiber Channel Connect relationship CIT to prevent multiple relationships between the same nodes from appearing in the top layer.



Storage Topology

This view maps logical dependencies between Logical Volumes on Hosts and Logical Volumes on Storage Arrays. There is no folding in this view.



Correlation Rules

The **Storage_Basic** package contains basic correlation rules to enable impact analysis and root cause analysis in UCMDB. These correlation rules are templates for more complex rules that you can define based on business needs.

All correlation rules fully propagate both Change and Operation events. For details on impact analysis, see "Impact Analysis Manager Page" and "Impact Analysis Manager Overview" in *Modeling Guide*.

To access the Storage_Basic package: **Administration > Package Manager**. For details, see "Package Manager" in *HP UCMDB Administration Guide*.

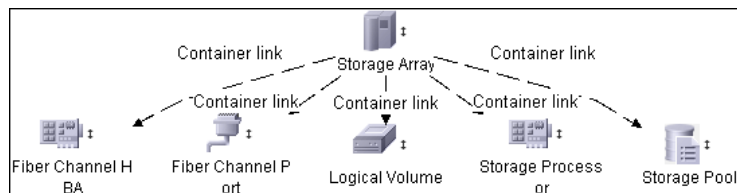
Note: Correlation events are not propagated to Fiber Channel Ports for performance reasons.

This section includes:

- "Storage Array Devices to Storage Array" on page 429
- "Host Devices to Host" on page 430
- "Logical Volume to Logical Volume" on page 430
- "FC Switch Devices to FC Switch" on page 430
- "FC Port to FC Port" on page 431

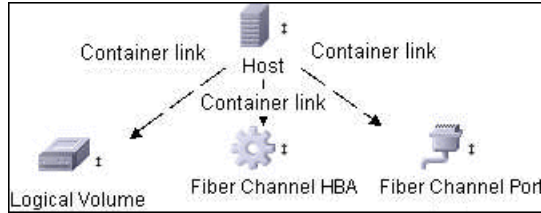
Storage Array Devices to Storage Array

This correlation rule propagates events between Logical Volumes, Storage Processors, Fiber Channel HBAs, and Storage Arrays.



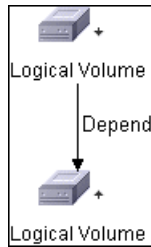
Host Devices to Host

This correlation rule propagates events between Fiber Channel HBAs and Hosts, and Logical Volumes on the Host.



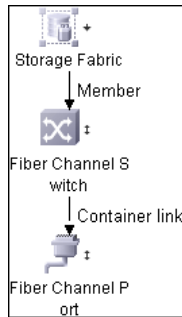
Logical Volume to Logical Volume

This correlation rule propagates events on a Logical Volume contained in a Storage Array to the dependent Logical Volume on the Host.



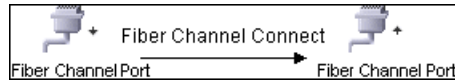
FC Switch Devices to FC Switch

This correlation rule propagates events from a Fiber Channel Port to and from a Switch. The event is also propagated to the associated Storage Fabric.



FC Port to FC Port

This rule propagates events on a Fiber Channel Port to another connected Channel Port.



Example Scenario of HBA Crashing on a Storage Array

- ▶ The event propagates from the HBA to the Storage Array and the Logical Volumes on the Array because of the Storage Devices to Storage Array rule.
- ▶ The correlation event on the Logical Volume then propagates to other dependent Logical Volumes through the Logical Volume to Logical Volume rule.
- ▶ Hosts using those dependent Logical volumes see the event next because of the Host Devices to Host rule.
- ▶ Depending on business needs, you define correlation rules to propagate events from these hosts to applications, business services, lines of business, and so on. This enables end-to-end mapping and impact analysis using UCMDB.

Reports

The **Storage_Basic** package contains basic reports that can be customized to suit the integrated ECC applications.

In addition to the system reports, Change Monitoring and Asset Data parameters are set on each CIT in this package, to enable Change and Asset Reports in UCMDB.

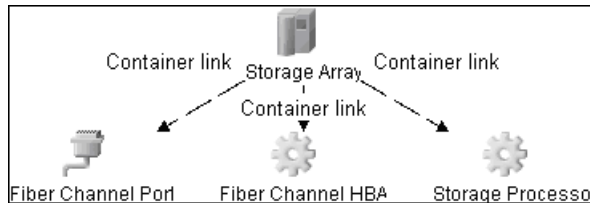
To access the Storage_Basic package: **Administration > Package Manager**. For details, see "Package Manager" in *HP UCMDB Administration Guide*.

This section includes:

- "Storage Array Configuration" on page 432
- "Host Configuration" on page 432
- "Storage Array Dependency" on page 433
- "Host Storage Dependency" on page 433
- "Storage Pool Configuration" on page 433

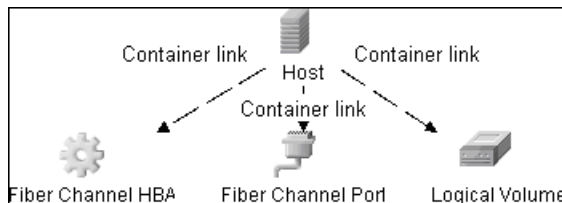
Storage Array Configuration

This report shows detailed information on Storage Arrays and its sub-components including Fiber Channel Ports, Fiber Channel Arrays, and Storage Processors. The report lists Storage Arrays with sub-components as children of the Array.



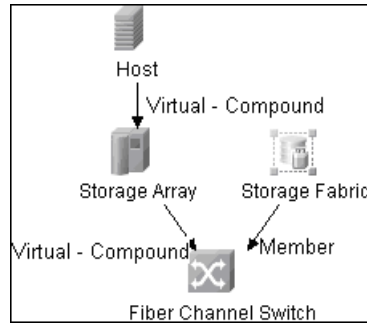
Host Configuration

This report shows detailed information on hosts that contain one or more Fiber Channel HBAs, Fiber Channel Ports, or Logical volumes. The report lists hosts with sub-components as children of the host.



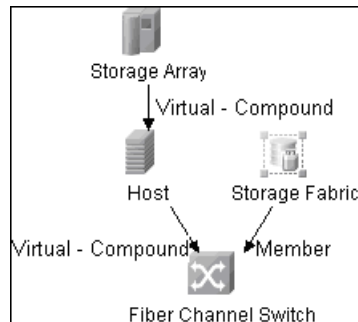
Storage Array Dependency

This report maps dependencies on a Storage Array. The report also displays information on switches connected to it.



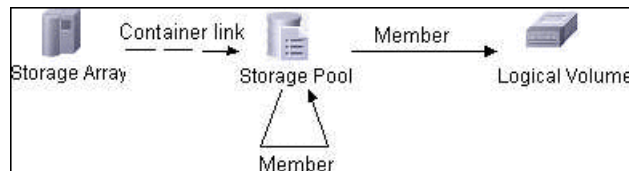
Host Storage Dependency

This report shows detailed information on storage infrastructure dependencies of a Host. The report lists hosts and dependent components.



Storage Pool Configuration

This report shows detailed information on Storage Pool configuration.



35

Data Dependency and Mapping Inventory Integration with HP Universal CMDB

This chapter includes:

Concepts

- Overview on page 436
- DDMi Adapter on page 436

Tasks

- Populate the CMDB with Data from DDMi on page 438
- Federate Data with DDMi on page 441
- Customize the Integration Data Model in UCMDB on page 441

Reference

- DDMi Adapter Configuration Files on page 443

Troubleshooting and Limitations on page 444

Concepts

Overview

This document describes how to integrate DDMi with UCMDB. Integration occurs by populating the UCMDB database with devices, topology, and hierarchy from DDMi and by federation with DDMi's supported classes and attributes. This enables change management and impact analysis across all business services mapped in UCMDB.

According to UCMDB reconciliation rules, if a CI is mapped to another CI in the CMDB, it is updated during reconciliation; otherwise, it is added to the CMDB.

Supported Versions

DDMi integration has been developed and tested on HP Universal CMDB version 7.5.2 or later with ED version 2.20 or DDMi version 7.5.

DDMi Adapter

Integration with DDMi is performed using a DDMi adapter, which is based on the Generic DB Adapter. This adapter supports full and differential population for defined CI types as well as federation for other CI types or attributes.

The DDMi adapter supports the following features:

- ▶ Full population of all instances of the selected CI Types.
- ▶ Identifying changes that have occurred in DDMi, to update them in UCMDB.
- ▶ Implementing **Remove** in DDMi. When a CI is removed in DDMi, it is not physically deleted from the database, but its status is changed to indicate that the CI is no longer valid. The DDMi adapter interprets this status as an instruction to remove the CI when needed.
- ▶ Federation of defined CI Types and attributes.

Out-of-the-box integration with DDMi includes population of the following classes:

- Node (some of the attributes are populated and some are federated)
- Layer2 connection
- Location that is connected to the node
- IP address
- Interface

In addition, the following classes can be defined as federated from DDMi:

- Asset
- CPU
- File system
- Installed software
- Printer
- Cost center

The following classes and attributes should be marked as federated by the DDMi adapter for the proper functionality of the Actual State feature of Service Manager:

- Classes
 - Person
 - Asset
 - CPU
 - Installed software
 - Printer
 - Windows service
- Node attributes
 - DiscoveredOsVendor
 - DiscoveredModel

- Description
 - DomainName
 - DiscoveredLocation
 - NetBiosName
-

Note: Avoid marking the **CreateTime** and **LastModifiedTime** attributes as federated, as it may lead to unexpected results.

Tasks

Populate the CMDB with Data from DDMi

This task describes how to install and use the DDMi adapter, and includes the following steps:

- "Define the DDMi integration" on page 438
- "Define a population job" on page 440
- "Run the population job" on page 440

1 Define the DDMi integration

- In UCMDB, navigate to **Data Flow Management > Integration Studio**.
- Click the **Create New Integration Point** button to open the New Integration Point Dialog Box. For details, see "Create New Integration Point/Edit Integration Properties Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.



Enter the following information:

Name	Recommended Value	Description
Adapter	DDMi	The type of the adapter that will be used to retrieve the external data from the DDMi database.
Credentials	<user defined>	Allows you to set credentials for integration points. For details, see "Domain Credential References" in the <i>HP Universal CMDB Data Flow Management Guide</i> .
DBName/SID	aggregate	The database name.
DB Type	MySQL	The type of database used by DDMi.
Hostname/IP	<user defined>	The name of the DDMi server.
Integration Description	<user defined>	Free text that describes the integration point.
Integration Name	<user defined>	The name you give to the integration point.
Is Integration Activated	selected	Select this checkbox to create an active integration point. You clear the checkbox if you want to deactivate an integration, for instance, to set up an integration point without actually connecting to a remote machine.
Port	8108	The port through which you access the DDMi database.
Probe Name	<user defined>	The name of the Data Flow Probe to be used.

- c Click **Test Connection** to verify the connectivity.

2 Define a population job

Select the Population tab to define a population job that uses the integration point you defined in step 1. For details, see "Create New Job Definition Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

Four out-of-the-box integration queries are provided:

- ▶ **hostDataImport** - use to import nodes. Imported data includes nodes whose NodeRole attribute is either null, or contains **desktop**, **server**, or **virtualized_system**. Nodes are identified either by their interface or IP address. Information also includes the location of the nodes (building, floor and room).
- ▶ **networkDataImport** - use to import nodes that are not imported with **hostDataImport**. Similar to **hostDataImport**, except that it imports nodes whose NodeRole is not null and does not contain the following strings: **desktop**, **server**, **virtualized_system**, or **printer**.
- ▶ **printerDataImport** - use to import printers. Similar to **networkDataImport**, except that it does import nodes whose NodeRole contains the string **printer**.
- ▶ **Layer2DataImport** - use to import Layer2 connections between pairs of nodes through their interfaces. Information also includes the nodes and their IP addresses.

3 Run the population job

Activate the population job in one of the following ways:



- ▶ To immediately run a full population job, click the **Run Full Job** button. In a full population job, all appropriate data is transferred, without taking the last run of the population job into consideration.



- ▶ To immediately run a differential population job, click the **Run Diff Job** button. In a differential population job, the previous population time stamp is sent to DDMi, and DDMi returns changes from that time stamp to the present. These changes are then entered into the UCMDB database.

- To schedule a differential population job to run at a later time or periodically, define a scheduled task. For details, see "Define Tasks that Are Activated on a Periodic Basis" in the *HP UCMDB Administration Guide*.

Federate Data with DDMi

The following steps describe how to define the CI Types that will be federated with DDMi.

- 1** In UCMDB, navigate to **Data Flow Management > Integration Studio**.
- 2** Select the integration point that you defined in step 1 on page 438.
- 3** Click the Federation tab. The panel shows the CI Types that are supported by the DDMi adapter.
- 4** Select the CI Types and attributes that you want to federate.
- 5** Click **Save**.



Customize the Integration Data Model in UCMDB

Out-of-the-box CIs for DDMi integration can be extended in one of the following ways:

To add an attribute to an existing CI type:

If the attribute you want to add does not already exist in the CMDB, you need to add it. For details, see "Add/Edit Attribute Dialog Box" in the *Modeling Guide*.

- 1** Navigate to the `orm.xml` file as follows: **Data Flow Management > Adapter Management > DDMiAdapter > Configuration Files > orm.xml**.
- 2** Locate the `generic_db_adapter.[CI type]` to be changed, and add the new attribute.

- 3** Ensure that the TQL queries that include this CI Type have the new attribute in their layouts, as follows:
 - a** In the Modeling Studio, right-click the node where you want to include the attribute.
 - b** Select **Query Node Properties**.
 - c** Click **Advanced layout settings** and select the new attribute.

For details about selecting attributes, see "Layout Settings Dialog Box" in the *Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 444.

To add a new CI Type to the DDMi Adapter:

- 1** In UCMDDB, create the CI Type that you want to add to the adapter, if it does not already exist. For details, see "Create a CI Type" in *Modeling Guide*.
- 2** Navigate to the **orm.xml** file as follows: **Data Flow Management > Adapter Management > DDMiAdapter > Configuration Files > orm.xml**.
- 3** Map the new CI type by adding a new entity called **generic_db_adapter.[CI type]**.
- 4** In the **orm.xml** file, ensure that the new CI Type has the following mappings:
 - ▶ the **data_note** attribute is mapped to the **NMID_StatusInAppliance** column (this attribute is used for checking the CI's status).
 - ▶ the **last_modified_time** and **create_time** attributes are mapped to the **Device_UpdatedDt** and **Device_FirstFoundDt** columns.

For details, see "The orm.xml File" in the *HP Universal CMDB Developer Reference Guide*.

- 5** Create queries to support the new CI Types that you added. Make sure that all mapped attributes have been selected in the Advanced Layout settings:
 - a** In the Modeling Studio, right-click the node where you want to include the attribute.
 - b** Select **Query Node Properties**.

- c Click **Advanced layout settings** and select the new attribute.

For details about selecting attributes, see "Layout Settings Dialog Box" in the *Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 444.

- 6 In UCMDB, navigate to **Data Flow Management > Integration Studio**.
- 7 Edit the DDMi integration point to support the new CI Type by selecting it either for population or for federation.
- 8 If the new CI Type is for population, edit the population job that you created in step 2 on page 440 to include the new TQL query.

Reference

DDMi Adapter Configuration Files

The adapter includes the following configuration files:

- **orm.xml**. The Object Relational mapping file in which you map between UCMDB classes and database tables.
- **discriminator.properties**. Maps each supported CI type (also used as a discriminator value in **orm.xml**) to a list of possible corresponding values of the discriminator column, **DeviceCategory_ID**.
- **replication_config.txt**. Contains a comma-separated list of non-root CI and relations types that have a **Remove** status condition in the DDMi database. This status condition indicates that the device has been marked for deletion.
- **fixed_values.txt**. Includes a fixed value for the attribute **ip_domain** in the class IP (**DefaultDomain**).

For details on adapter configuration, see "Generic Database Adapter" in the *HP Universal CMDB Developer Reference Guide*.

Troubleshooting and Limitations

Note: Only queries that meet these requirements are visible to the user when selecting a query for a population job.

- ▶ Queries that are used in population jobs should contain one CI Type that is labeled with a **Root** prefix, or one or more relations that are labeled with a **Root** prefix.

The root node is the main CI that is synchronized; the other nodes are the contained CIs of the main CI. For example, when synchronizing the **Node** CI Type, that graph node is labeled as **Root** and the resources are not labeled **Root**.

- ▶ The TQL graph must not have cycles.
- ▶ A query that is used to synchronize relations should have the cardinality 1...* and an OR condition between the relations.
- ▶ The adapter does not support compound relations.
- ▶ The TQL graph should contain only CI types and relations that are supported by the DDMi adapter.
- ▶ ID conditions on the integration TQL query are not supported.

36

Microsoft SCCM/SMS Integration with HP Universal CMDB

This chapter includes:

Concepts

- SCCM/SMS Integration – Overview on page 446
- SMS Adapter on page 447

Tasks

- Populate the CMDB with Data from SCCM/SMS on page 449
- Federate Data with SCCM/SMS on page 453
- Customize the Integration Data Model in Universal CMDB on page 454

Reference

- SCCM/SMS Integration Package on page 455
- SMS Adapter Configuration Files on page 458

Troubleshooting and Limitations on page 459

Concepts

SCCM/SMS Integration – Overview

This document includes the main concepts, tasks, and reference information for integration of Microsoft System Center Configuration Manager (SCCM)/Systems Management Server (SMS) with HP Universal CMDB.

Integration occurs by populating the UCMDB database with devices, topology, and hierarchy from SCCM/SMS and by federation with SCCM/SMS supported classes and attributes.

According to UCMDB reconciliation rules, if a CI (in SCCM/SMS) is already mapped to a CI in the CMDB, it is updated; otherwise, it is added to the CMDB.

Microsoft System Center Configuration Manager/Systems Management Server are used by IT administrators to manage client computers and servers.

SCCM/SMS enable you to:

- ▶ manage computers that roam from one location to another
- ▶ track deployment and use of software assets, and use this information to plan software procurement and licensing
- ▶ provide IT administrators and management with access to data accumulated by SCCM/SMS
- ▶ provide scalable hardware and software management
- ▶ manage security on computers running Windows operating systems, with a minimal level of administrative overhead

This section also includes:

- "Supported Versions" on page 447

Supported Versions

Integration has been developed and tested on HP Universal CMDB version 8.03 or later, with SCCM version 2007 or SMS version 2003.

SMS Adapter

Integration with SCCM/SMS is performed using an SMS adapter, which is based on the Generic DB Adapter. This adapter supports full and differential population for defined CI types as well as federation for other CI types or attributes.

The SMS Adapter supports the following features:

- Full replicating of all instances of the selected CI types.
- Identifying changes that have occurred in SCCM/SMS, to update them in the UCMDB.
- Simulating the touch mechanism capabilities:

When a CI is removed from SCCM/SMS, it is physically deleted from the database and there is no way to report about it. The SMS Adapter supports a full synchronization interval. This means that the adapter transfers data for which the aging mechanism has been enabled, and provides the time interval to run a full synchronization that simulates the touch mechanism.

- Federation of selected CI types and attributes.

Out-of-the-box integration with SCCM/SMS includes population of the following classes:

- Node (some of the attributes are populated and some are federated)
- Layer2 connection
- Location that is connected to the node
- IP address

- Interface

In addition, the following classes can be defined as federated from SCCM/SMS:

- CPU
- File system
- Installed software
- Windows service

The following classes and attributes should be marked as federated by the SCCM/SMS adapter for the proper functionality of the Actual State feature of Service Manager:

- Classes
 - CPU
 - Installed software
 - Windows service
- Node attributes
 - DiscoveredOsVendor
 - DiscoveredModel
 - Description
 - DomainName
 - NetBiosName

Note: Avoid marking the **LastModifiedTime** attribute as federated, as it may lead to unexpected results.

Tasks

Populate the CMDB with Data from SCCM/SMS

This task describes how to install and use the SMS adapter.

This task includes the following steps:

- "Define the SMS Integration" on page 449
- "Define a Population Job" on page 450
- "Run the population job" on page 452

1 Define the SMS Integration

a Navigate to **Data Flow Management > Integration Studio**.



b Click the **Create New Integration Point** button to open the New Integration Point Dialog Box. For details, see "Create New Integration Point/Edit Integration Properties Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

Enter the following information:

Name	Recommended Value	Description
Adapter	Microsoft SMS	The type of the adapter that will be used to retrieve the external data from the SCCM/SMS database.
Credentials	<user defined>	Allows you to set credentials for integration points. For details, see "Domain Credential References" in the <i>HP Universal CMDB Data Flow Management Guide</i> .
DB Type	SQL Server	The type of database used by SCCM/SMS
DBName/SID	smsbpn	The database name.

Name	Recommended Value	Description
Hostname/IP	<user defined>	The host name of the machine where the database of SCCM/SMS is running.
Integration Description	<user defined>	A description of the integration point.
Integration Name	<user defined>	The name you assign to the integration point.
Is Integration Activated	selected	Select this checkbox to create an active integration point. You clear the checkbox if you want to deactivate an integration, for instance, to set up an integration point without actually connecting to a remote machine.
Port	1433	The port through which you access the MSSQL database.
Probe Name	<user defined>	The name of the Data Flow Probe to be used.

- c Click **Test Connection** to verify the connectivity.

2 Define a Population Job

Select the Population tab to define one or more population jobs that use the integration point you defined above. For details, see "Create New Job Definition Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

The following integration queries are provided out of the box:

- **hostDataImport**. Imports nodes. Imported data includes nodes whose **NodeRole** attribute is either null, or contains the string desktop, server, or virtualized_system. Nodes are identified either by their interface or IP address. Information also includes the location of the nodes (building, floor and room).

- **networkDataImport.** Imports snodes that are not imported with hostDataImport. Similar to hostDataImport, except that it imports nodes whose NodeRole is not null and does not contain the strings desktop, server, virtualized_system, or printer.
- **printerDataImport.** Imports printers. Similar to networkDataImport , except that it does import nodes whose NodeRole contains the string printer.
- **Layer2DataImport.** Imports Layer2 connections between pairs of nodes through their interfaces. Information also includes the nodes and their IP addresses.

Data removal issue

The SMS Adapter cannot supply information about removed CIs, since the Microsoft SMS database does not contain this information. The integration uses the population mechanism to simulate the touch mechanism. This behavior is achieved by performing a one-time full synchronization from SCCM/SMS to UCMDB followed by similar scheduled synchronizations. CI items that did not update during a synchronization are candidates for deletion by the aging mechanism and will be deleted once the aging period has elapsed.

The SMS Adapter provides a mechanism that enables you to set a time frame during which to run the synchronization (the default time frame is one week). This mechanism works with a scheduled population job to run full population only (not differential population) according to the time frame value you set.

To change the time frame, navigate to **Data Flow Management > Adapter Manager > SMS Adapter > Adapters > SMSAdapter**. Right-click **SMS Adapter** and select **Edit Adapter Source**. Modify the value of the **full-population-days-interval** field and click **Save**.

Notes:

- ▶ This integration assumes that the aging mechanism in CMDB is active.
 - ▶ If no scheduled population with SCCM/SMS is defined, the data that has been populated from SCCM/SMS will eventually be removed from the CMDB, since the touch mechanism will recognize that no changes have been made.
 - ▶ Since the deletion of CIs is performed by the aging mechanism, the **Allow Deletion** check box in the Population Job definition is irrelevant. CIs are always deleted if not touched by the aging mechanism.
-

3 Run the population job

Activate the population job in one of the following ways:



- ▶ To immediately run a full population job, click the **Run Full Job** button. In a full population job, all appropriate data is transferred, without taking the last run of the population job into consideration.



- ▶ To immediately run a differential population job, click the **Run Diff Job** button. In a differential population job, the previous population time stamp is sent to SCCM/SMS, and SCCM/SMS returns changes from that time stamp to the present. These changes are then entered into the UCMDB database.
- ▶ To schedule a differential population job to run at a later time or periodically, define a scheduled task. For details, see "Define Tasks that Are Activated on a Periodic Basis" in the *HP UCMDB Administration Guide*.

Note that the replicated CIs are controlled by the integration TQL that is used. You can create additional TQL queries that contain different topologies for use in other jobs.

Federate Data with SCCM/SMS

The following steps describe how to define the CI types that will be federated with SCCM/SMS.

- 1** In UCMDB, navigate to **Data Flow Management > Integration Studio**.
- 2** Select the integration point that you defined in step 1 on page 449.
- 3** Click the Federation tab. The panel shows the CI types that are supported by the SMS adapter.
- 4** Select the CI types and attributes that you want to federate.
- 5** Click **Save**.

Note:

- ▶ CI types that populate UCMDB should not be selected for federation. Specifically, avoid federating node, IP address, interface, location, and Layer2, which populate UCMDB out-of-the-box.
 - ▶ Other CI types can be used in federation only after the node data has been replicated to CMDB by the hostDataImport TQL query. This is because the default reconciliation rule is based on node identification.
-

Customize the Integration Data Model in Universal CMDB

Out-of-the-box CIs for SCCM/SMS integration can be extended in one of the following ways:

To add an attribute to an existing CI type:

If the attribute you want to add does not already exist in the CMDB, you need to add it. For details, see "Add/Edit Attribute Dialog Box" in the *Modeling Guide*.

- 1** Navigate to the `orm.xml` file as follows: **Data Flow Management > Adapter Management > SMS Adapter > Configuration Files > orm.xml**.
- 2** Locate the `generic_db_adapter.[CI type]` to be changed, and add the new attribute.
- 3** Ensure that the TQL queries that include this CI type have the new attribute in their layouts as follows:
 - a** In the Modeling Studio, right-click the node where you want to include the attribute.
 - b** Select **Query Node Properties**.
 - c** Click **Advanced Layout Settings** and select the new attribute.

For details about selecting attributes, see "Layout Settings Dialog Box" in the *Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 459.

To add a new CI Type to the Generic DB Adapter:

- 1** In UCMDB, create the CI Type that you want to add to the adapter, if it does not already exist. For details, see "Create a CI Type" in the *Modeling Guide*.
- 2** Navigate to the `orm.xml` file as follows: **Data Flow Management > Adapter Management > SMS Adapter > Configuration Files > orm.xml**.
- 3** Map the new CI type by adding a new entity called `generic_db_adapter.[CI type]`.

For more details, see "The orm.xml File" in the *HP Universal CMDB Developer Reference Guide*.

- 4 Create queries to support the new CI types that you have added. Make sure that all mapped attributes are selected in the Advanced Layout settings:
 - a In the Modeling Studio, right-click the node where you want to include the attribute.
 - b Select **Query Node Properties**.
 - c Click **Advanced layout settings** and select the new attribute.

For details about selecting attributes, see "Layout Settings Dialog Box" in *Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 459.

- 5 In UCMDB, navigate to **Data Flow Management > Integration Studio**.
- 6 Edit the SMS integration point to support the new CI type by selecting it either for population or for federation.
- 7 If the new CI type is for population, edit the population job that you created above.

Reference

SCCM/SMS Integration Package

This section includes:

- "Transformations" on page 456
- "SCCM/SMS Plug-in" on page 457
- "Reconciliation" on page 458

Transformations

Following is the list of transformations that are applied to values when they are transferred to or from the SCCM/SMS database:

CMDB Class	Attribute	Transformation
windows	nt_servicepack	<p>Represents number of the Windows service pack.</p> <p>SCCM/SMS DB: Service Pack 2</p> <p>UCMDB: 2.0</p> <p>Transformer: standard GenericEnumTransformer, mapped in the nt.nt_servicepack.transformer.xml file.</p>
node	host_isdesktop	<p>A Boolean value that determines whether a machine is a desktop or a server.</p> <p>SCCM/SMS DB: Workstation or Server</p> <p>UCMDB: true or false</p> <p>Transformer: standard GenericEnumTransformer, mapped in the node.host_isdesktop.transformer.xml file.</p>
node	host_os	<p>Represents the node's operation system.</p> <p>SCCM/SMS DB. Microsoft Windows XP Professional</p> <p>UCMDB. Windows XP</p> <p>Transformer. Standard GenericEnumTransformer, mapped in the node.discovered_os_name.transformer.xml file.</p> <p>If the SCCM/SMS operation system value is not listed in the transformer.xml file, the original value is sent to UCMDB.</p> <p>By default, only Windows operating systems are mapped.</p>

CMDB Class	Attribute	Transformation
node	host_osinstalltype	Represents the Windows OS edition. SCCM/SMS DB. Microsoft Windows XP Professional UCMDB. Professional Transformer. Standard GenericEnumTransformer , mapped in the host.host_osinstalltype.transformer.xml file. Note: The same column in the SCCM/SMS database is mapped to two different UCMDB attributes, using different transformers.
disk device	data_name	Represents the partition name. SCCM/SMS DB. C: UCMDB. C Transformer. standard AdapterToCmdbRemoveSuffixTransformer that removes the colon.
interface	interface_macaddr	Represents the MAC address of NIC. SCCM/SMS DB. AB:CD:EF:01:23:45 UCMDB. ABCDEF012345 Transformer. custom SmsMacAddressTransformer that removes the colons from the SCCM/SMS MAC address while making it compatible with the UCMDB MAC addresses.

SCCM/SMS Plug-in

The **SmsReplicationPlugin** provides enhanced functions to those found in the Generic Database Adapter. It is called when:

- full topology is requested (**getFullTopology**) – this returns all the CIs that were found in the external SCCM/SMS database.
- topology layout is requested (**getLayout**)

- ▶ topology of changes is requested (**getChangesTopology**) – this returns only the CIs that are modified or added after a specific time. The topology of the changes is calculated as follows:
 - ▶ There is a specific date (**fromDate**) after which all changes are requested.
 - ▶ Most of the entities in the SCCM/SMS database contain a Timestamp column that contains the date and time of the last modification. This Timestamp column is mapped to the **root_updatetime** attribute of a CI. Currently, some entities do not contain any creation time information. The entities that have a timestamp column must be listed in the **replication_config.txt** file.
 - ▶ In the integration TQL query, the node CI is named **Root**.
 - ▶ Using the plug-in, the integration TQL query is dynamically modified so that each **Root** entity and all entities that are listed in the **replication_config.txt** file have an additional condition causing the value of the **root_updatetime** attribute to be greater than or equal to the **fromDate** value.
 - ▶ This modified TQL query is then used to obtain the data.

Reconciliation

The adapter uses the default reconciliation rule-based mapping engine.

SMS Adapter Configuration Files

The adapter includes the following configuration files:

- ▶ **orm.xml**. The Object Relational mapping file, which maps between SCCM/SMS database tables and columns, and UCMDB classes and attributes. Both CIs and links are mapped.
- ▶ **fixed_values.txt**. Used by the Generic DB Adapter to set the **ip_domain** of IP Address CIs to DefaultDomain.
- ▶ **plugins.txt**. Contains configuration information for the Generic DB Adapter. Also defines three plug-ins that are used during replication: **getFullTopology**, **getChangesTopology**, and **getLayout**.

- **transformations.txt**. Contains the configuration for transformation of attribute values. For a list of the transformations, see "Transformations" on page 456.
- **node.discovered_os_name.transformer.xml**. Mapping used by the transformer for the **host_isdesktop** attribute.
- **node.host_osinstalltype.transformer.xml**. Mapping used by the transformer for the **host_os** attribute.
- **host.host_osinstalltype.transformer.xml**. Mapping used by the transformer for the **host_osinstalltype** attribute.
- **nt.nt_servicepack.transformer.xml**. Mapping used by the transformer for the **nt_servicepack** attribute.
- **replication_config.txt**. Contains a comma-separated list of non-root CIs and relations types that have a **timestamp** condition in the SCCM/SMS database. This status condition indicates the last time the entity was updated.
- **reconciliation_types.txt**. Defines the CI types that are used for reconciliation.

For details on adapter configuration, see "Generic Database Adapter" in the *HP Universal CMDB Developer Reference Guide*.

Troubleshooting and Limitations

- Queries that are used in population jobs should contain one CI type that is labeled with a Root prefix, or one or more relations that are labeled with a Root prefix.

The root node is the main CI that is synchronized; the other nodes are the contained CIs of the main CI. For example, when synchronizing the Node CI Type, that graph node is labeled as Root and the resources are not labeled Root.

- The TQL graph must not have cycles.
- A query that is used to synchronize relations should have the cardinality 1...* and an OR condition between the relations.
- The adapter does not support compound relations.

- ▶ Entities that are added in SCCM/SMS are sent as updates to UCMDB by the SMS Adapter during differential population.
- ▶ ID conditions on the integration TQL query are not supported.
- ▶ The TQL graph should contain only CI types and relations that are supported by the SCCM/SMS adapter.

Index

A

- Active Directory
 - discover 107
 - discover domain controllers, topology 108
- Active Directory Connection by LDAP job 110
- Active Directory Topology by LDAP job 111
- adsutil.vbs 31
- Alteon application switch by SNMP job 41
- Apache Tomcat by Shell job 33, 348, 350
- Application – SAP Solution Manager
 - module 157
- architecture 371

B

- Books Online 14
- Bugzilla, Wordpress, MediaWiki 353

C

- Cisco CSS by SNMP job 42
- Class C IPs by ICMP job 166
- configuration files
 - SMS DB Adapter 458
- conventions, typographical 16
- credential-less discovery 255
- CSV data
 - importing from external source 314

D

- Data Flow Probe
 - copying files to a remote Windows machine 30
- Databases TCP Ports job 74, 79

- DB2 Connection by SQL job 74
- DB2 Topology by SQL job 74
- DDMi integration 435
 - overview 436
- discovery
 - credential-less 255
 - IBM DB2 Server 73
 - IIS 355
 - JBoss 195
 - JBoss by Shell 198
 - Layer 2 277
 - Microsoft Cluster Server 49
 - Microsoft Exchange Server 2007 122
 - Microsoft Load Balancing 51
 - Microsoft SQL Server 77
 - network - basic 233
 - Oracle 93
 - SAP 151
 - SAP Solution Manager 157
 - Siebel 163
 - Solaris zones 219
 - Veritas Cluster Server 67
 - VMware 321
 - WebLogic 203
 - WebLogic by JMX 203
 - WebSphere 211
 - WebSphere by Shell 214
 - WebSphere Connections by JMX 212
- Discovery Tools 295
- diskinfo.exe 32
- DNS Zone by Nslookup job 230
- documentation updates 16
- documentation, online 14

E

- ECC 413

Index

- discovering storage topology 415
- overview 414
- SQL queries for job 422
- ECC Integration by SQL job 418
- EMC Control Center
 - overview 414
- EMC Control Center integration 413
 - discovering storage topology 415
 - SQL queries for job 422
- Exchange_Server_2007_Discovery.ps1 33
- external sources
 - importing data from 297
 - importing data, troubleshooting 318

F

- F5 BIG-IP LTM by SNMP job 41
- File Monitor by Shell
 - limitation 296
- File Monitor by Shell job 33
- F-Secure
 - running on Windows for Host Connection by Shell job 244

G

- general information 19, 29
- GetFileModificationDate.vbs 33
- getfilever.vbs 31

H

- Host Connection by Shell job 30, 85, 123, 154, 220, 350, 360
 - Windows running F-Secure 244
- Host Connection by Shell/WMI/SNMP job in ECC 417
- Host Connection by SNMP job 41, 220
- Host Connection by WMI job 117, 166, 220
- Host Fingerprint using nmap job 256
- Host Networking By SNMP job 279
- host resources
 - and application dependency, overview 263
 - and applications discovery 263
 - troubleshooting and limitations 275
 - workflow 266

- Host Resources and Applications by Shell job 30, 31, 81, 123, 154, 162, 220, 263, 270, 350, 360
- Host Resources and Applications by Shell/WMI/SNMP job
 - in ECC 417
- Host Resources and Applications by SNMP job 220, 270
- Host Resources and Applications by WMI job 81, 220, 270, 274
- Host Resources and Applications module 263
- HP Software Support Web site 15
- HP Software Web site 16

I

- IBM DB2 Server
 - discovery 73
- IHS Websphere Plugin by Shell job 33
- IIS
 - discovery 355
- IIS Applications by NTCMD job 31, 33, 356, 360
- import data from external sources 301
- Import from CSV File job 300, 301
- Import from Database job 300
- Import from Properties file job 300
- importing data
 - from external sources 297
 - troubleshooting 318
- installed software
 - reverting to previous discovery method 274
- integration package
 - SMS 455
- integrations
 - with ECC 415
- IP Traffic by Network Data job 224

J

- J2EE JBoss by Shell job 197, 199
- J2EE TCP Ports job 213
- J2EE WebLogic by Shell job 208
- J2EE Weblogic by Shell job 33
- J2EE WebSphere by Shell job 34, 216

- J2EE WebSphere by Shell or JMX job 33, 213
- J2EE WebSphere Connections by JMX job 213
- JBoss
 - discovery 195
 - discovery by Shell 198
- JBoss by JMX job 197
- JBoss Connections by JMX job 197
- jobs
 - Alteon application switch by SNMP 41
 - Apache Tomcat by Shell 348, 350
 - Cisco CSS by SNMP 42
 - Class C IPs by ICMP 166
 - Databases TCP Ports 74, 79
 - DB2 Connection by SQL 74
 - DB2 Topology by SQL 74
 - DNS Zone by Nslookup 230
 - F5 BIG-IP LTM by SNMP 41
 - Host Connection by Shell 85, 123, 154, 220, 350, 360
 - Host Connection by SNMP 41, 220
 - Host Connection by WMI 117, 166, 220
 - Host Fingerprint using nmap 256
 - Host Networking By SNMP 279
 - Host Resources and Applications by Shell 81, 123, 154, 162, 220, 263, 270, 350, 360
 - Host Resources and Applications by SNMP 220, 270
 - Host Resources and Applications by WMI 81, 220, 270
 - IIS Applications by NTCMD 356, 360
 - Import from CSV File 300, 301
 - Import from Database 300
 - Import from Properties 300
 - IP Traffic by Network Data 224
 - J2EE JBoss by Shell 197, 199
 - J2EE TCP Ports 213
 - J2EE WebLogic by Shell 208
 - J2EE WebSphere by Shell 216
 - J2EE WebSphere by Shell or JMX 213
 - J2EE WebSphere Connections by JMX 213
 - JBoss by JMX 197
 - JBoss Connections by JMX 197
 - Layer2 Topology Bridge based by SNMP 284
 - Layer2 Topology VLAN based by SNMP 287
 - Microsoft Exchange Connection by NTCMD 123
 - Microsoft Exchange Connection by WMI 115, 117, 120
 - Microsoft Exchange Topology by NTCMD 123
 - Microsoft Exchange Topology by WMI 115, 117, 120
 - MS Cluster by NTCMD 50
 - MSSQL Connection by SQL 79
 - MSSQL Topology by SQL 79
 - MySQL by Shell 84, 85
 - Oracle Listeners by Shell 97
 - Oracle RAC Topology by Shell 98
 - Potential Servers by Network Data 225
 - Range IPs by ICMP 154, 220, 350
 - SAP ABAP Connection by SAP JCO 155
 - SAP ABAP Topology by SAP JCO 155
 - SAP Applications by SAP JCO 155
 - SAP ITS by NTCMD 155
 - SAP Java Topology by SAP JMX 162
 - SAP Solution Manager by SAP JCO 155
 - SAP System by Shell 154, 159, 162
 - SAP TCP Ports 159
 - SE Integration by SQL 396
 - Server Ports by Network Data 224
 - Servers by Network Data 223
 - Siebel DB by TTY 166, 170
 - Siebel DB by WMI and NTCMD 166
 - Siebel Web Applications by NTCMD 166
 - Siebel Web Applications by TTY 166
 - Software Element CF by Shell 271
 - TCP Ports 166
 - Veritas Cluster by Shell 68
 - VLAN ports by SNMP 282
 - VLANS by SNMP 280
 - VMware ESX Connection by VIM 326
 - VMware ESX Topology by VIM 326

Index

- VMware VirtualCenter Connection by WMI and VIM 326
- VMware VirtualCenter Topology by VIM 326
- WebLogic by Shell 206
- WebServer Detection using TCP Ports 155, 166
- WebServices by URL 356
- junction.exe 34

K

- Knowledge Base 15

L

- Layer 2
 - discovery 277
- Layer2 Topology Bridge based by SNMP job 284
- Layer2 Topology VLAN based by SNMP job 287

M

- meminfo.exe 32
- Microsoft Cluster Server
 - discovery 49
- Microsoft Exchange Connection by NTCMD job 33, 123
- Microsoft Exchange Connection by WMI job 115, 117, 120
- Microsoft Exchange Server
 - discover 116
 - discover topology with Active Directory 125
 - overview 107, 115
- Microsoft Exchange Server 2007
 - discovery 122
 - package 124, 130
- Microsoft Exchange Topology by LDAP job 130
- Microsoft Exchange Topology by NTCMD job 33, 123
- Microsoft Exchange Topology by WMI job 115, 117, 120
- Microsoft Internet Information Services (IIS)

355

- Microsoft Message Queue
 - discovery 135
 - methodology 139
 - new entities 149
 - task 135
- Microsoft Network Load Balancing
 - discovery 51
- Microsoft SQL Server
 - discovery 77
- MS Cluster by NTCMD job 50
- MSSQL Connection by SQL job 79
- MSSQL Topology by SQL job 79
- MySQL by Shell job 84, 85

N

- network - basic 233
- NLB 51
- NNMi integration 369
 - change management and impact analysis 384, 385
 - connection protocol parameters 385, 389
 - overview 370
 - run 373
 - troubleshooting and limitations 390
- NNMi-UCMDB Integration 381

O

- online documentation 14
- Online Help 15
- online resources 15
- Oracle
 - discovery 93
- Oracle Listeners by Shell job 97
- Oracle RAC Topology by Shell job 98
- Oracle TNSName by Shell job 34
- OS credentials
 - discovery for MS SQL Server 77
- Overview
 - DDMi integration 436

P

- Potential Servers by Network Data job 225

processlist.exe 32

R

Range IPs by ICMP job 154, 220, 350
in ECC 417

Readme 14

reg_mam.exe 31

Running Software CF by Shell job 34

S

SAP

discover ABAP 152

discover Java 160

discovery 151

SAP ABAP Connection by SAP JCO job 155

SAP ABAP discovery overview 152

SAP ABAP Topology by SAP JCO job 155

SAP Applications by SAP JCO job 155

SAP ITS by NTCMD job 155

SAP Java Topology by SAP JMX job 162

SAP Profiles by Shell job 34

SAP Solution Manager

discovery 157

SAP Solution Manager by SAP JCO job 155

SAP System By Shell job 34

SAP System by Shell job 154, 159, 162

SAP TCP Ports job 159

SE Integration by SQL job 396

Server Ports by Network Data job 224

Servers by Network Data job 223

Service Guard Cluster Topology by TTY job
34

Siebel

discovery 163

Siebel Application Server Configuration job
34

Siebel DB by TTY job 166, 170

Siebel DB by WMI and NTCMD job 166

Siebel Web Applications by NTCMD job 166

Siebel Web Applications by TTY job 166

SMS

integration package 455

SMS DB Adapter configuration files 458

SMS integration

overview 446

SMS supported versions 447

SmsDbAdapter 447

software

reverting to previous discovery

method 274

Software Element CF by Shell job 271

Solaris zones

discovery 219

SQL Server

shallow discovery 80

shallow discovery overview 81, 83,
348

Storage Essentials (SE)

integration with Universal CMDB 393

supported versions

SMS 447

T

TCP Ports job 166

TempWmicBatchFile.bat

empty file created 30

troubleshooting

SAP discovery fails 162

Troubleshooting and Knowledge Base 15

typographical conventions 16

U

UCMDB

DDMi integration 435

UDDI

discover processes 172

updates, documentation 16

V

Veritas Cluster by Shell job 34, 68

Veritas Cluster Server

discovery 67

VLAN ports by SNMP job 282

VLANS by SNMP job 280

VMware

discovery 321

VMware ESX Connection by VIM job 326

VMware ESX Topology by VIM job 326

Index

- VMware VirtualCenter Connection by WMI and VIM job 326
- VMware VirtualCenter Topology by VIM job 326
- VMware VMotion
 - discover 343
- VMware VMotion Monitor by VIM job 345

W

- WebLogic
 - discovery 203
 - discovery by JMX 203
- WebLogic by Shell job 206
- Webserver by Shell job 34
- WebServer Detection using TCP Ports job 155, 166
- WebServices by URL job 356
- WebSphere
 - discovery 211
 - discovery by Shell 214
 - troubleshooting and limitations 217
- WebSphere Connections
 - discovery by JMX 212
- What's New 14

X

- xCmdSvc.exe
 - connecting to remote Windows machine 30