

HP Universal CMDB

for the Windows and Linux operating systems

Software Version: CP7.00, 9.02

Discovery and Integration Content Guide

Document Release Date: October 2010

Software Release Date: October 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	15
How This Guide Is Organized	15
Who Should Read This Guide	16
HP Universal CMDB Online Documentation	16
Additional Online Resources.....	19
Documentation Updates	20

PART I: INTRODUCTION

Chapter 1: Supported Content	23
Discovered Applications.....	24
Discovered Operating Systems	34
Windows Localized Versions.....	34
Supported Integration	35
Chapter 2: General Information for Discovery and Integration Content	37
Database Connections by SQL Jobs	38
Delete Files Copied to Remote Machine	41
Files Copied to a Remote Machine.....	42
Troubleshooting and Limitations	47

PART II: DISCOVERY CONTENT

Chapter 3: Load Balancers	51
Overview.....	52
Discover Load Balancers.....	53
Chapter 4: High Availability Cluster Multiprocessing (HACMP)	61
Overview.....	62
Discover IBM HACMP	63
Discovery Mechanism	70
Chapter 5: Microsoft Cluster.....	77
Discover Microsoft Cluster Servers.....	78

Chapter 6: Microsoft Network Load Balancing (NLB)	81
Microsoft NLB Overview	82
Discovery Mechanism	82
Discovering NLB with the Command Line Utility	83
Discover Microsoft Network Load Balancing Systems.....	86
MS NLB Cluster CIT	91
NLB Cluster Software CIT.....	92
Configuration Document (NLB Port Rule)	93
Glossary	94
Components of the Network Load Balancing Architecture	95
Chapter 7: Sun Cluster	97
Overview.....	98
Discover Sun Cluster	99
Discovery Mechanism	104
Chapter 8: Veritas	127
Discover Veritas Cluster Servers	128
Chapter 9: Database Connections by Host Credentials	131
Overview.....	132
Discovery Mechanism	132
Discover Database Connections by Host Credentials.....	134
Chapter 10: DB2	143
Discover IBM DB2 Databases	144
Chapter 11: MS-SQL	147
Discovery by OS Credentials	148
Discover Microsoft SQL Server Database Application.....	149
Discover SQL Server by OS Credentials.....	152
Chapter 12: MySQL Replication Between Databases	155
Overview.....	156
Discover MySQL Configuration and Replication Jobs.....	157
Chapter 13: Oracle	165
Discover Oracle Databases.....	166
Discover Oracle Real Application Cluster (RAC).....	168
Chapter 14: Active Directory	181
Overview.....	182
Discover Active Directory Domain Controllers and Topology.....	183

Chapter 15: Microsoft Exchange	191
Overview.....	192
Discover Microsoft Exchange Server 2003	193
Discover Microsoft Exchange Server 2007	199
Discover Microsoft Exchange Server Topology with Active Directory	202
Chapter 16: Microsoft MQ (Message Queue)	211
Discover Microsoft MQ	212
Topology Discovery Methodology	216
Added Entities	226
Removed Entities.....	227
Chapter 17: SAP	229
SAP Discovery Overview.....	230
Discover SAP ABAP	233
Discover SAP Solution Manager	238
Discover SAP Java	241
Troubleshooting and Limitations	245
Chapter 18: Siebel	247
Overview.....	248
Discover Siebel Topology	249
Troubleshooting and Limitations	260
Chapter 19: UDDI Registry	261
Overview.....	262
Discover UDDI Processes.....	263
Chapter 20: WebSphere MQ	265
Overview.....	266
Discover WebSphere MQ	269
Discovered CITs.....	273
Relationships	276
Enrichment Rule.....	279
Views and Reports	280
Troubleshooting and Limitations	281
Chapter 21: JBoss	283
JBoss Discovery Overview	284
Discover JBoss by JMX.....	285
Discover JBoss by Shell.....	287
Troubleshooting and Limitations	291

Chapter 22: WebLogic	293
Discover J2EE WebLogic by JMX	294
Discover J2EE WebLogic by Shell.....	298
Troubleshooting and Limitations	302
Chapter 23: WebSphere.....	303
WebSphere Discovery Overview	304
Discover WebSphere by JMX	305
Discover WebSphere by Shell.....	309
Troubleshooting and Limitations	312
Chapter 24: Active and Passive Discovery Network Connections ...	315
Overview.....	316
Discover Processes	317
Discover TCP Traffic.....	324
Chapter 25: Network Basic.....	327
Network – Basic Overview	328
Network Workflow Overview.....	328
Discover Host Connection by Shell	329
Discover Host Connection by SNMP	332
Discover Host Connection by WMI.....	335
Discover Windows Running F-Secure with the Host Connection by Shell Job.....	339
Windows Processes.....	340
UNIX-Based Processes	342
Chapter 26: Credential-less.....	349
Overview.....	350
Discover Host Fingerprint with Nmap.....	351
Chapter 27: Network – DNS.....	357
Overview.....	358
Discover DNS Zone by Nslookup	360
Discover DNS Zone by DNS	363
Discovery Mechanism – Windows	366
Discovery Mechanism – UNIX-like	367
Glossary	369
Chapter 28: Host Resources and Applications.....	371
Host Resources and Applications Overview.....	372
Discover Host Resources and Applications	375
Revert to Previous Method of Discovering Installed Software	381
Troubleshooting and Limitations	381

Chapter 29: Layer 2	383
Overview.....	384
Discover Layer 2 Objects	385
Chapter 30: Discovery Tools	399
Overview.....	400
Troubleshooting and Limitations	400
Chapter 31: Import from Excel Workbook	401
Discover Import from Excel Workbook	402
Troubleshooting and Limitations	405
Chapter 32: Importing Data from External Sources	407
Importing Data from External Sources Overview	408
The External_source_import Package.....	410
The Import from CSV File Job.....	411
The Import from Database Job.....	415
The Import from Properties File Job.....	419
The External Source Mapping Files	421
Convert Strings to Numbers.....	422
Import CSV Data from an External Source – Scenario.....	424
Troubleshooting and Limitations	428
Chapter 33: HP Partitioning Solution	431
Overview.....	432
Discover HP vPars and nPars.....	433
Views.....	437
Discovery Mechanism	440
Troubleshooting and Limitations	469
Chapter 34: IBM HMC	471
Overview.....	472
Discover IBM HMC.....	473
The IBM HMC by Shell Job	480
The IBM LPar and VIO by Shell Job	480
The IBM_HMC_SHELL_PATTERN Adapter	481
The IBM_LPAR_VIO_BY_SHELL Adapter	482
Discovery Mechanism	483
VIO Server Side Commands	498
LPAR Side Commands.....	510
Troubleshooting and Limitations	511

Chapter 35: Hyper-V	513
Overview.....	514
Discover Hyper-V	515
Discovery Mechanism	521
Troubleshooting and Limitations	528
Chapter 36: Solaris Zones	529
Overview.....	530
Discover Solaris Zones	531
The Solaris Zones_by_TTY Job	535
Discovery Mechanism	536
Troubleshooting and Limitations	550
Chapter 37: VMware	551
Discover VMware Infrastructure Topology	552
Discover VMware VMotion	571
Troubleshooting and Limitations	575
Chapter 38: XEN.....	577
Overview.....	578
Discover Xen	579
Discovery Mechanism	587
Chapter 39: Apache Tomcat	593
Overview.....	594
Discover Apache Tomcat.....	596
Discover Bugzilla, Wordpress, and MediaWiki	600
Chapter 40: Microsoft Internet Information Services (IIS)	603
Discover Microsoft Internet Information Services (IIS) – Previous Topology.....	604
Discover Microsoft Internet Information Services (IIS) – Current Topology.....	607

PART III: SUPPORTED INTEGRATIONS

Chapter 41: HP ServiceCenter/Service Manager Integration	617
Adapter Usage.....	619
Supported Versions.....	620
Data Push Flow	620
Federation Use Cases	621
Viewing the Actual State	622
The serviceDeskConfiguration.xml File	625
Deploy the Adapter – Typical Deployment.....	634
Deploy the ServiceDesk Adapter	634
Add an Attribute to the ServiceCenter/Service Manager CIT	640
Communicate with Service Manager over SSL	647
Set Up Service Manager Integration for Data Push.....	648
Add a New Attribute to an Existing CI Type.....	651
Add a New CI Type.....	652
Flow and Configuration	654
Troubleshooting and Limitations	661
Chapter 42: Network Node Manager i (NNMi) Integration	665
NNMi Integration Overview	666
NNMi - UCMDB Integration Architecture	668
Set Up HP NNMi–HP UCMDB Integration	669
Run HP NNMi–UCMDB Integration	670
Use the HP NNMi–HP UCMDB Integration	678
Change the HP NNMi–HP UCMDB Integration Configuration	681
Disable HP NNMi–HP UCMDB Integration Configuration	681
Perform Impact Analysis	683
HP NNMi–HP UCMDB Integration Configuration Form Reference	684
Troubleshooting and Limitations	688
Chapter 43: Storage Essentials (SE) Integration with HP Universal CMDB	691
SE Integration – Overview	692
Discover the SE Oracle Database	693
Storage Essentials Integration Packages	695
Discovered CITs.....	695
Views.....	700
Impact Analysis Rules.....	704
Reports.....	706

Chapter 44: HP Systems Insight Manager (HP SIM) Integration.....	709
Overview.....	710
Discovery Mechanism	710
Discover HP SIM Data Center Infrastructure	714
Instance Views.....	723
Troubleshooting and Limitations	725
Chapter 45: EMC Control Center (ECC) Integration with HP Universal CMDB	727
ECC Integration – Overview	728
Discover the ECC Storage Topology	729
ECC Job SQL Queries.....	737
Views	739
Impact Analysis Rules.....	744
Reports.....	747
Chapter 46: Data Dependency and Mapping Inventory Integration with HP Universal CMDB.....	751
Overview.....	752
DDMi Adapter	753
Populate the CMDB with Data from DDMi.....	755
Federate Data with DDMi.....	758
Customize the Integration Data Model in UCMDB	758
DDMi Adapter Configuration Files	761
Troubleshooting and Limitations	762
Chapter 47: Microsoft SCCM/SMS Integration with HP Universal CMDB	763
SCCM/SMS Integration – Overview	764
SMS Adapter	765
Populate the CMDB with Data from SCCM/SMS	767
Federate Data with SCCM/SMS	771
Customize the Integration Data Model in UCMDB	772
SCCM/SMS Integration Package.....	774
SMS Adapter Configuration Files	777
Troubleshooting and Limitations	778
Chapter 48: Atrium Push Adapter.....	781
Overview.....	782
Integrate UCMDB with Remedy or Atrium.....	783
Integration Mechanism	789
Mapping Files	789
Troubleshooting and Limitations	794
Index.....	795

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 15
- ▶ Who Should Read This Guide on page 16
- ▶ HP Universal CMDB Online Documentation on page 16
- ▶ Additional Online Resources on page 19
- ▶ Documentation Updates on page 20

How This Guide Is Organized

The guide contains the following chapters:

Part I Introduction

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

Part II Discovery Content

Describes how to discover system components.

Part III Supported Integrations

Describes how to retrieve data by integration with other systems.

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:

- the *HP Universal CMDB Deployment Guide* PDF. 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.
- the *HP Universal CMDB Database Guide* PDF. Explains how to set up the database (MS SQL Server or Oracle) needed by HP Universal CMDB.

- ▶ the *HP Universal CMDB Discovery and Integration Content Guide* PDF. 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 repositories 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 repositories and how to set up HP Universal CMDB to discover network components.
- ▶ **UCMDB 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).



Topic Types

Within this guide, each subject area is organized into topics. A topic contains a distinct module of information for a subject. The topics are generally classified according to the type of information they contain.

This structure is designed to create easier access to specific information by dividing the documentation into the different types of information you may need at different times.

Three main topic types are in use: **Concepts**, **Tasks**, and **Reference**. The topic types are differentiated visually using icons.

Topic Type	Description	Usage
Concepts 	Background, descriptive, or conceptual information.	Learn general information about what a feature does.
Tasks 	<p>Instructional Tasks. Step-by-step guidance to help you work with the application and accomplish your goals. Some task steps include examples, using sample data. Task steps can be with or without numbering:</p> <ul style="list-style-type: none"> ▶ Numbered steps. Tasks that are performed by following each step in consecutive order. ▶ Non-numbered steps. A list of self-contained operations that you can perform in any order. 	<ul style="list-style-type: none"> ▶ Learn about the overall workflow of a task. ▶ Follow the steps listed in a numbered task to complete a task. ▶ Perform independent operations by completing steps in a non-numbered task.
	<p>Use-case Scenario Tasks. Examples of how to perform a task for a specific situation.</p>	Learn how a task could be performed in a realistic scenario.

Topic Type	Description	Usage
 Reference	General Reference. Detailed lists and explanations of reference-oriented material.	Look up a specific piece of reference information relevant to a particular context.
	User Interface Reference. Specialized reference topics that describe a particular user interface in detail. Selecting Help on this page from the Help menu in the product generally open the user interface topics.	Look up specific information about what to enter or how to use one or more specific user interface elements, such as a window, dialog box, or wizard.
 Troubleshooting and Limitations	Troubleshooting and Limitations. Specialized reference topics that describe commonly encountered problems and their solutions, and list limitations of a feature or product area.	Increase your awareness of important issues before working with a feature, or if you encounter usability problems in the software.

Additional Online Resources

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

HP Software Support accesses the HP Software Support Web site. This site 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/hpsoftwaresupport.

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 accesses the HP Software Web site. This 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 24
- ▶ Discovered Operating Systems on page 34
- ▶ Windows Localized Versions on page 34
- ▶ Supported Integration on page 35

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, 5.5, 6.0	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, 10	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...
BMC	Atrium CMDB	1.1, 2.0, 2.1, 7.5, 7.6	Remedy	Pushes configuration items (CIs) from HP UCMDB to the Atrium CMDB server using mapping xml files. Note: Synchronized Content, not discovery of application topology
BMC	Remedy ARS	6.3, 7.0, 7.1, 7.5, 7.6	Remedy	Pushes configuration items (CIs) from HP UCMDB to Remedy ARS using mapping xml files. Note: Synchronized Content, not discovery of application topology
Cisco	CSS	6.10, 7.4	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 Note: Cisco WebNS is the software version running on the 11000 and 11500 series CSS
Citrix	XEN	3.4	SSH, Telnet	Bridge, CPU, Execution Environment, File System, File System Export, Interface, Layer2Connection, Node, Physical Port, Virtualization Layer Software, Xen domain config

Vendor	Product	Versions	Credentials	Discovers...
EMC	EMC Control Center (ECC)	6.0.1	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	Mapping of Virtual IPs to real IP addresses of servers configured for load balancing; configuration files, load balancing algorithms, and end user IP addresses
HP	Network Node Manager (NNM)	8.1, 9.0, 9.1	NNM API	Discovered nodes, IPs, networks, interfaces and Layer 2 connection information to create a Layer 2 topology in UCMDB
HP	nPartitions	A.03xx, A.04xx, A.05xx	SSH, Telnet	CPU, Fibre Channel HBA, File System, HP Complex, HP nPar Config, HP vPar Config, I/O Chassis, CellBoard, Interface, nodes, Physical Volume, SCSI Adapter, Volume Group

Vendor	Product	Versions	Credentials	Discovers...
HP	ServiceGuard		Shell	SG cluster software, SG packages, SG resources, cluster members
HP	SIM	5.1, 5.2, 5.3, 6.0, 6.1	HP SIM	<p>Synchronized configuration items (CIs) include nodes such as Windows, and UNIX servers, network devices, printers, clusters, cellular/partitioned systems, blade enclosures, and racks. Some server components, for example, CPU, are also synchronized. The integration also synchronizes relationships between blade servers and blade enclosures, virtual machines, physical servers, and so on.</p> <p>Note: Synchronized Content, not discovery of application topology</p>
HP	Storage Essentials (SE)	6.0.0	SQL	<p>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</p>

Vendor	Product	Versions	Credentials	Discovers...
IBM	DB2 Universal Database (UDB)	8.2, 9.1, 9.5, 9.7	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>
IBM	HACMP	5.4	SSH, Telnet	<p>Topology (configured networks, node interfaces—both public TCP/IP and serial heartbeat, and service IPs) and Application Resources (configured resource groups, application servers, and volume groups)</p>
IBM	HMC	4, 5	SSH, Telnet	<p>CPU, I/O Slot, IBM Frame, IBM HMC, IBM LPar Profile, IBM Processor Pool, Interface, Node, Virtualization Layer Software, SCSI Adapter, Physical Port, Physical Volume, Fibre Channel HBA, File System, SEA Adapter</p>
IBM	HTTP Server	5, 6.1, 7	Shell	<p>IBM Http Server's WebSphere plug-in configuration by parsing the IHS plug-in configuration file</p>

Vendor	Product	Versions	Credentials	Discovers...
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, 6.1, 7.0	Shell	J2EE Server, J2EE application, JDBC datasource, Database, EJB Module, Web Module, J2EE Domain and JMS resources
JBoss	Application Server	3.x, 4.0, 4.2, 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
Microsoft	Active Directory	2003	LDAP	Domain Controllers, Forest, Domain, Site, and Trusts

Chapter 1 • Supported Content

Vendor	Product	Versions	Credentials	Discovers...
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, Windows Server 2008	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	Hyper-V	Windows 2008, Windows 2008 R2	NTCMD, WMI	Resource pools, virtual switches, virtual NICs, virtual machines, and configuration files
Microsoft	IIS	6,7	Shell	Discover the IIS Web Server, IIS Web Site, IIS virtual Dir, IIS Application pool, web services and configuration files

Vendor	Product	Versions	Credentials	Discovers...
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?
MySQL	MySQL Database	3.x, 4.x, 5.0, 5.1	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

Vendor	Product	Versions	Credentials	Discovers...
Oracle	Siebel CRM	7.5, 7.7, 8.0, 8.1	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
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

Vendor	Product	Versions	Credentials	Discovers...
Sun	Solaris Cluster	3.2	SSH, Telnet	Cluster Software, Configuration file, Execution Environment, Node, Sun Cluster, Sun Cluster Resource, Sun Resource Group
Sun	Solaris Zones	5.1	Shell	Containers, zones, and share resources
Sybase	Adaptive Server Enterprise	10.x, 11.x, 12.x, 15.0, 15.5	SQL	Sybase databases, including instances, tablespaces, users, processes, jobs (backup routines, log routines, and so on), and any database objects
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, 4	Shell	
VMware	ESX & ESXi	2.5, 3, 3i, 3.5, 4	VIM	ESX servers, cluster groups, virtual resource groups
VMware	vCenter (formerly Virtual Center)	2.01, 2.5, 4	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, Local Users
HP	HP-UX	10.xx, 11.xx		OS, Memory, Disks, CPU, Processes, Software (packages), Services (Daemons), Files, 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, Local Users
Sun	Solaris	5.9, 5.10		OS, Memory, Disks, CPU, Processes, Software (packages), Services (daemons), Files, Local Users
Microsoft	Windows	All Versions		OS, Memory, Disks, CPU, Processes, Software, Services, Files, Local Users

Windows Localized Versions

Discovery is supported for the following localized versions of Windows:

- ▶ Chinese
- ▶ French
- ▶ German
- ▶ Italian
- ▶ Japanese
- ▶ Korean
- ▶ Portuguese
- ▶ Russian

- Spanish

Supported Integration

- HP ServiceCenter/Service Manager
- Network Node Manager i (NNMi)
- Storage Essentials (SE)
- HP Systems Insight Manager (HP SIM)
- EMC Control Center (ECC)
- Data Dependency and Mapping Inventory
- Microsoft SCCM/SMS
- Atrium Push Adapter

2

General Information for Discovery and Integration Content

This chapter includes:

Concepts

- ▶ Database Connections by SQL Jobs on page 38

Tasks

- ▶ Delete Files Copied to Remote Machine on page 41

Reference

- ▶ Files Copied to a Remote Machine on page 42

Troubleshooting and Limitations on page 47

Concepts

Database Connections by SQL Jobs

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

To enable consistency across database connection jobs, so that all jobs follow the same workflow, changes have been made to the database modules. Each job uses its own adapter and trigger query, but a single script (**SQL_Connection.py**) is used for all jobs, for connection and reporting.

This alignment of connections by SQL across all database modules means that the SQL credentials do not have to include a predefined port or Oracle/DB2 SID to be able to connect to the database.

This section includes the following topics:

- "New Jobs" on page 38
- "Removed Jobs" on page 39
- "Trigger Query Changes" on page 39
- "Input Query Changes" on page 39
- "Triggered CI Data" on page 40
- "Adapter Parameters" on page 40

New Jobs

- DB2 Universal Database Connection by SQL
- MSSQL Server Connection by SQL
- MySQL Connection by SQL
- Oracle Database Connection by SQL

- Sybase Database Connection by SQL

Removed Jobs

These jobs have been moved to <<Legacy>> > Database > Connection by SQL.

- DB2 Connection by SQL
- MSSQL Connection by SQL
- MSSQL Server Credentials by SQL
- Oracle Connection by SQL
- Oracle Credentials by SQL
- Sybase Connection by SQL

Trigger Query Changes

The Trigger query includes **Node** with a connected IP address, **dbserver**, and **IpServiceEndpoint** with the following cardinalities:

```
Containment (Node, IpAddress) : 1..* AND (Composition (Node, IpServiceEndpoint) : 1..* OR Composition (Node, DbServer) : 1..*)
```

DbServer must be of the appropriate type (for example, **Sybase** or **Oracle**); **IpServiceEndpoint** must have the appropriate name (for example, **Sybase** or **Oracle**).

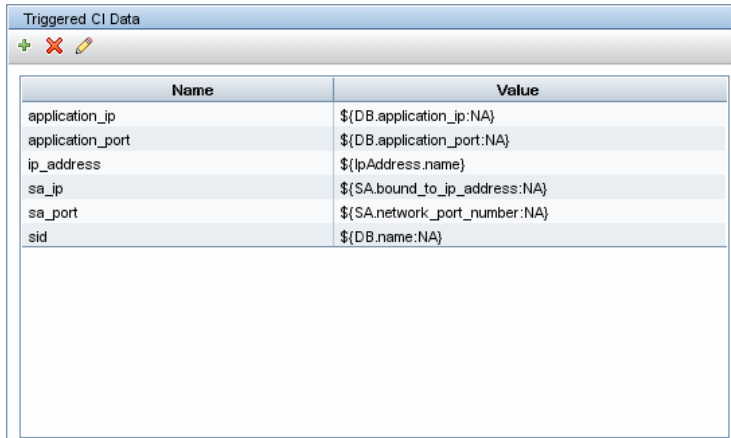
Input Query Changes

The Input query is similar to the Trigger query, but the cardinality is different:

```
Containment (SOURCE, IpAddress) : 1..* AND Composition (SOURCE, SA) : 0..* AND Composition (SOURCE, DB) : 0..*
```

Triggered CI Data

The `SQL_Connection.py` script adds information to a database CI according to the Triggered CI data values:

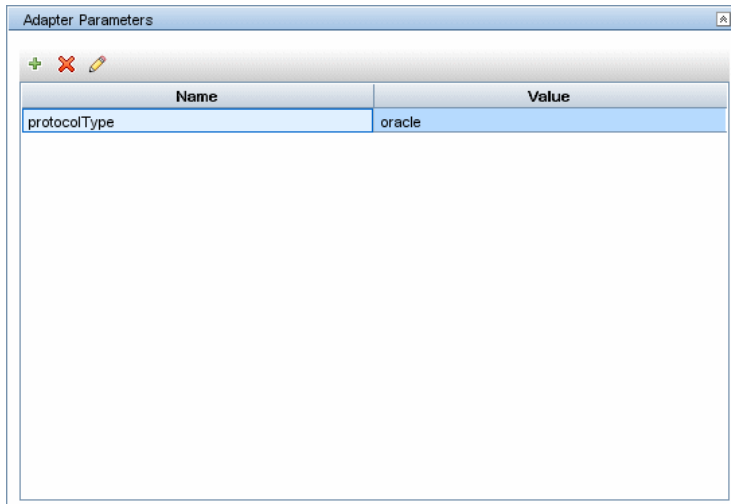


The screenshot shows a window titled "Triggered CI Data" with a table containing the following data:

Name	Value
application_ip	\${DB.application_ip:NA}
application_port	\${DB.application_port:NA}
ip_address	\${ipAddress.name}
sa_ip	\${SA.bound_to_ip_address:NA}
sa_port	\${SA.network_port_number:NA}
sid	\${DB.name:NA}

Adapter Parameters

The value of the `protocolType` adapter parameter defines which database is being discovered:



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

Name	Value
protocolType	oracle

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 42.

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.

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

Note:

- ▶ Data Flow 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 43
- ▶ "getfilever.vbs" on page 43
- ▶ "reg_mam.exe" on page 43
- ▶ "meminfo.exe" on page 44
- ▶ "diskinfo.exe" on page 44
- ▶ "processlist.exe" on page 44
- ▶ "Exchange_Server_2007_Discovery.ps1" on page 45
- ▶ "GetFileModificationDate.vbs" on page 45
- ▶ "junction.exe" on page 46

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,node</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 52

Tasks

- ▶ Discover Load Balancers on page 53

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 53
- "Prerequisites" on page 54
- "Discovery Workflow" on page 55
- "Load Balancer CITs" on page 55
- "The Load_balancing Package" on page 57
- "The Alteon_application_switch Package" on page 57
- "The F5_BIGIP_LTM Package" on page 58
- "The Cisco_CSS Package" on page 59
- "Topology Map" on page 60

1 Supported Versions

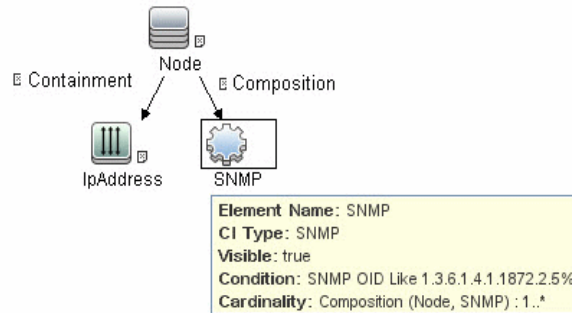
The supported version for each load balancer is as follows:

- 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

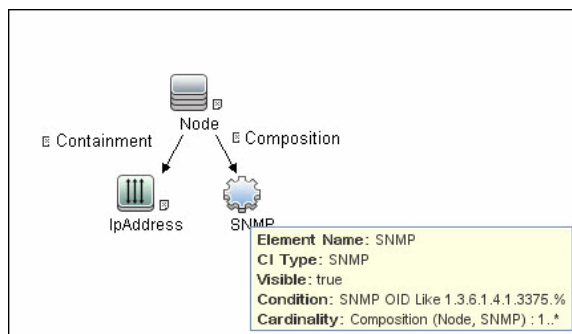
Run the **Host Connection by SNMP** job to discover and create SNMP CIs which answer the following requirements:

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



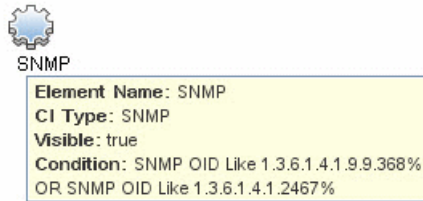
SNMP OID Like 1.3.6.1.4.1.1872.2.5%

- To be the trigger query 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 query 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%

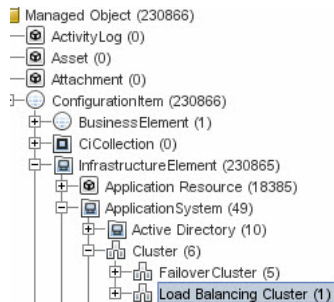
3 Discovery Workflow

- **Host Connection by SNMP.** For details on the prerequisites to running a load balancer job, see "Prerequisites" on page 54.
- Any of the following jobs:
 - **F5 BIG-IP LTM by SNMP**
 - **Alteon application switch by SNMP**
 - **Cisco CSS by SNMP**

4 Load Balancer CITs

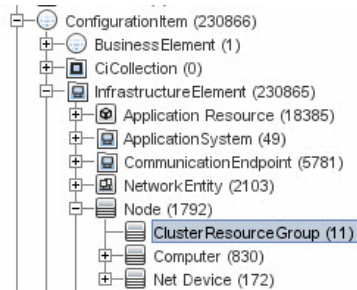
The following CITs model load balancer topology:

Load Balancer Software:



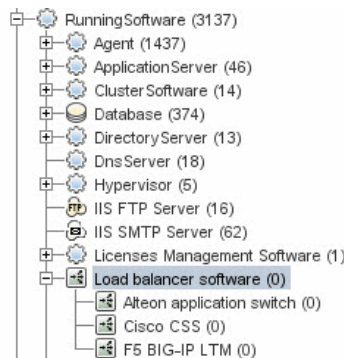
This CIT represents software that provides load balancing solutions. For details on the supported load balancers, see "Overview" on page 52.

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.

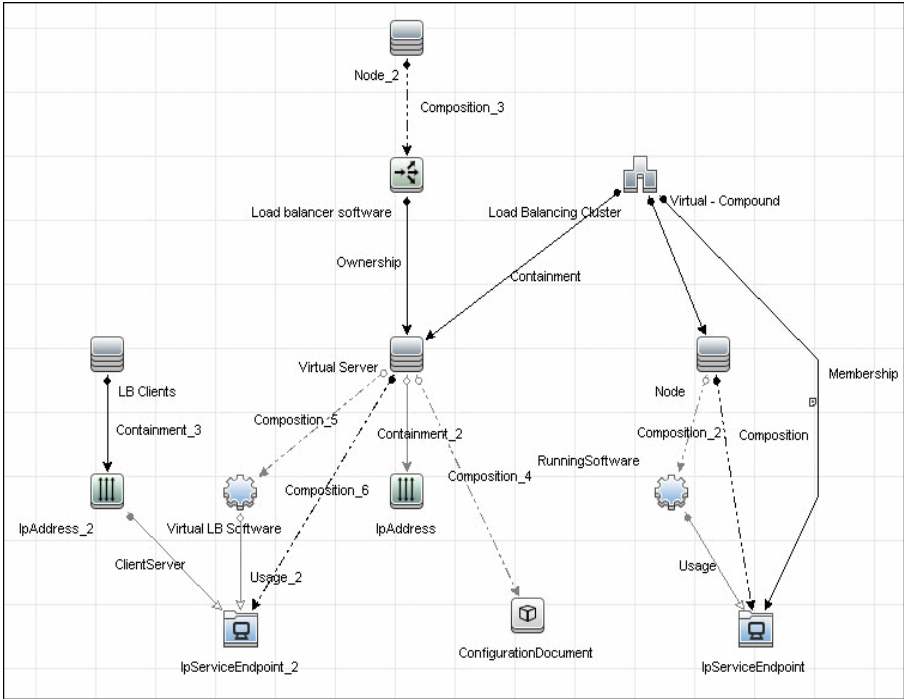
Load Balancing Cluster



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 the Load Balancer Topology view (**Modeling Studio > 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	ltmVirtualServTable	1.3.6.1.4.1.3375.2.2.10.1.2.1
Pools	ltmPoolTable	1.3.6.1.4.1.3375.2.2.5.1.2.1
Pools to server	ltmVirtualServPool Table	1.3.6.1.4.1.3375.2.2.10.6.2.1
Pool members	ltmPoolMemberTable	1.3.6.1.4.1.3375.2.2.5.3.2.1

Table Name	Name From MIB	OID
Rules to servers	ItmVirtualServRule Table	1.3.6.1.4.1.3375.2.2.10.8.2.1
Rules	ItmRuleTable	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):

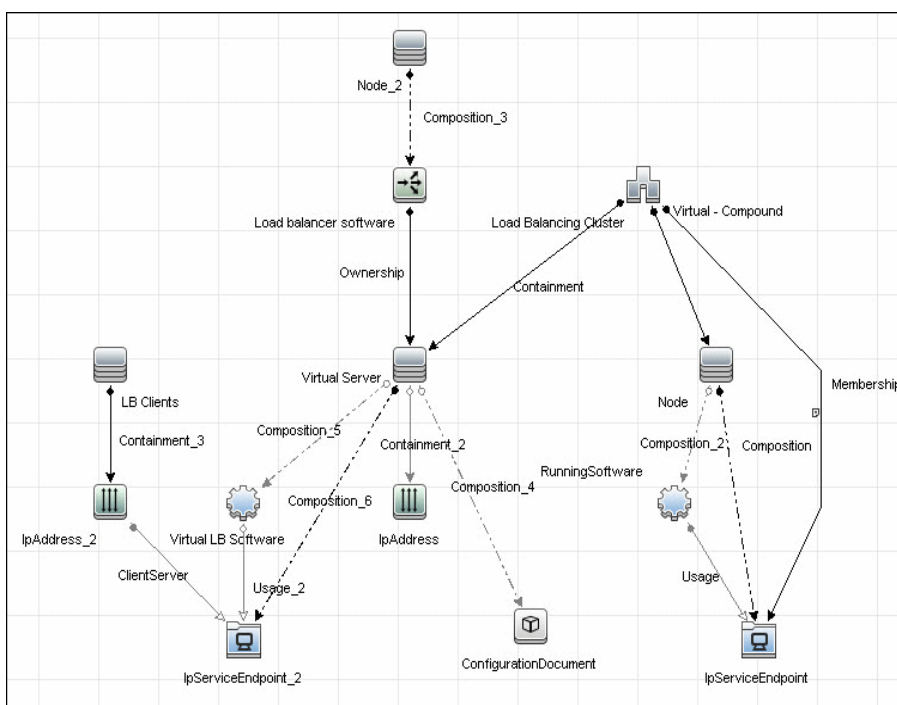
- **apCntTable** provides information about virtual addresses, virtual services, and pools.
- **apSvcTable** provides information about physical hosts included in the pool.

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

apSvcTable can contain entries for which there is no corresponding row in **apCntsvcTable**. In this case, such hosts are skipped.

Table name	Name from MIB	OID
CNT	apCntTable	1.3.6.1.4.1.2467.1.16.4.1 or 1.3.6.1.4.1.9.9.3681.16.4.1
SVC	apSvcTable	1.3.6.1.4.1.2467.1.15.2.1 or 1.3.6.1.4.1.9.9.3681.15.2.1
CNT to SVC	apCntsvcEntry	1.3.6.1.4.1.2467.1.18.2.1 or 1.3.6.1.4.1.9.9.3681.18.2.1

9 Topology Map



4

High Availability Cluster Multiprocessing (HACMP)

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 62

Tasks

- ▶ Discover IBM HACMP on page 63

Reference

- ▶ Discovery Mechanism on page 70

Concepts

Overview

High Availability Cluster Multiprocessing (HACMP) is an IBM solution for high-availability clusters on the AIX UNIX and Linux for IBM System p platforms.

HACMP can run on up to 32 computers or nodes, each of which is either actively running an application (active) or waiting to take over should another node fail (passive). Data on file systems can be shared between systems in the cluster.

HACMP relies heavily on IBM's Reliable Scalable Cluster Technology (RSCT). RSCT includes daemons which are responsible for monitoring the state of the cluster (for example, a node, NIC or network crash) and for coordinating the response to these events. HACMP is an RSCT aware client. RSCT is distributed with AIX.

The **IBM_HACMP** package discovers HACMP on AIX via TTY (SSH or Telnet protocols). The package follows the discovery model to discover the HACMP Topology (configured networks, node interfaces-both public TCP/IP and serial heartbeat, and service IPs) and Application Resources (configured resource groups, application servers, and volume groups). The package maps the configured public interfaces to UCMDDB IPs, serial interfaces to directories beneath the UCMDDB hosts, as well as volume groups to logical disks beneath the UCMDDB host, and Application Resources to the Topology.

Tasks

Discover IBM HACMP

This task includes the following steps:

- "Supported Version" on page 63
- "Prerequisites" on page 63
- "Class Model" on page 64
- "Instance View" on page 64
- "HACMP Topology Discovery Trigger Query (Shell not NTCMD HACMP)" on page 65
- "HACMP Application Discovery Trigger Query (Shell in HACMP Cluster)" on page 65
- "HACMP Application Discovery Input Query" on page 66
- "Discovery Workflow" on page 66
- "Created/Changed Entities" on page 68

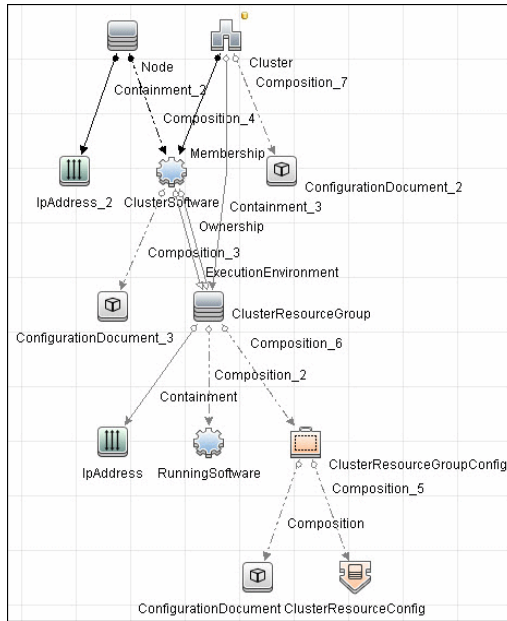
1 Supported Version

HACMP 5.4 on AIX 5.3

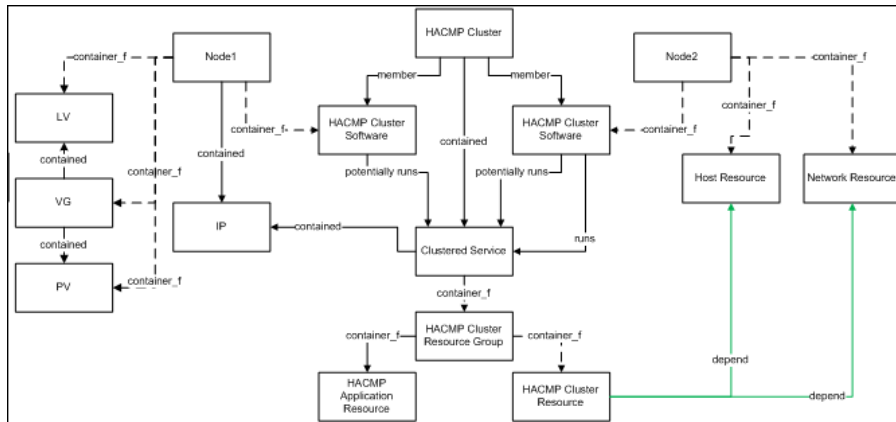
2 Prerequisites

- Verify that the Host Connection adapters have been successfully run on the nodes involved in the cluster:
 - For details, see "Network Basic" on page 327.
- To use the Shell protocols, configure the appropriate credentials.
 - For credentials information, see "SSH Protocol" and "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- Load the Storage Topology add-on package prior to deployment of the HACMP package.

3 Class Model

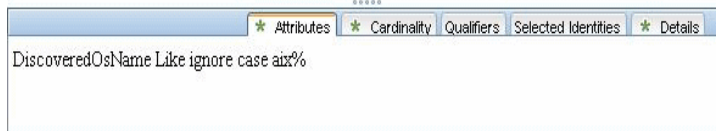
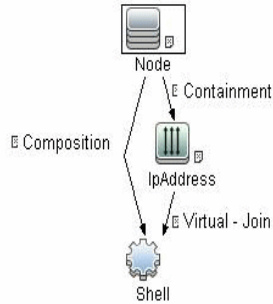


4 Instance View

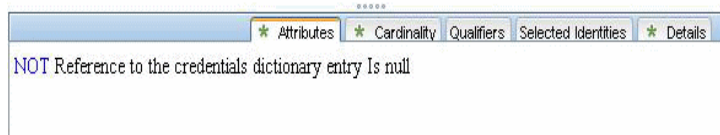
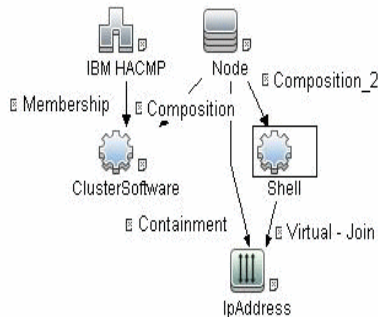


5 HACMP Topology Discovery Trigger Query (Shell not NTCMD HACMP)

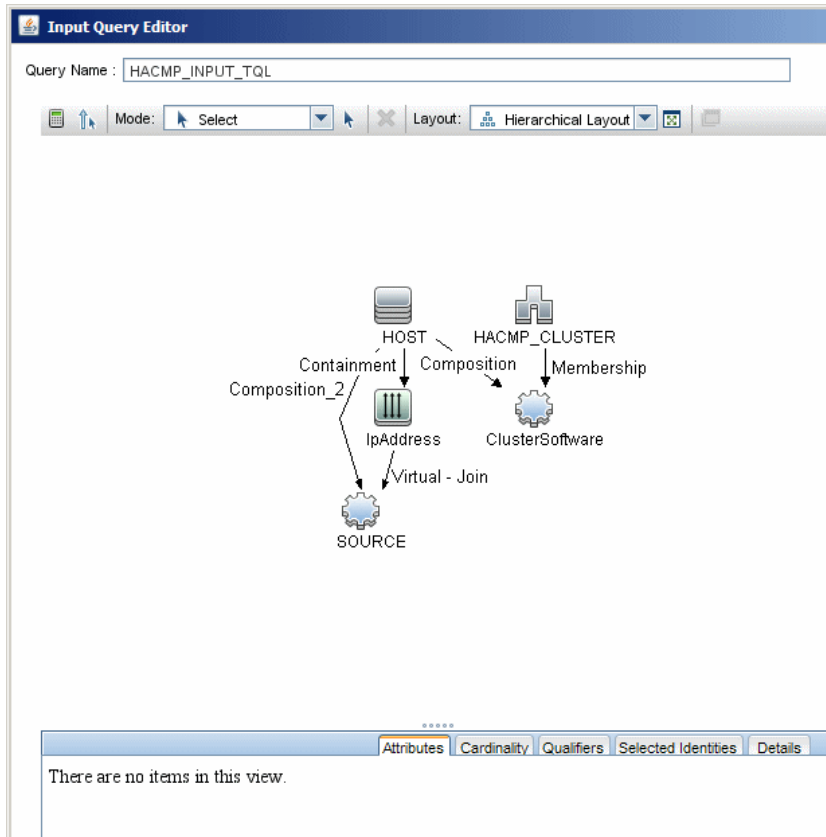
This trigger requires a TTY Shell that is not an NTCMD Shell.



6 HACMP Application Discovery Trigger Query (Shell in HACMP Cluster)



7 HACMP Application Discovery Input Query



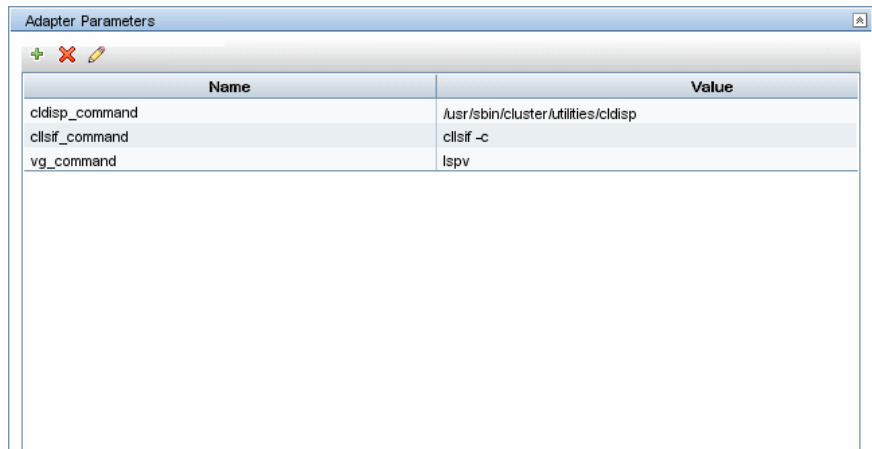
8 Discovery Workflow

For details on jobs, see "Discovery Control Panel – Advanced Mode Workflow" in *HP Universal CMDB Data Flow Management Guide*.

- a** Verify that the Probe has an IP range assigned to it that includes the IPs of the target machines running IBM HACMP Cluster.
- b** Verify that the Shell (SSH or Telnet) credentials are specified. For details, see "Prerequisites" on page 63.
- c** Run the **Range IPs by ICMP** job to discover which of the machines in the IP range are up.

- d** Run the **Host Connection by Shell** job to discover Shell connectivity and basic information about the hosts.
- e** Verify that the **Host Connection** jobs have previously discovered the hosts that are to be part of the HACMP cluster. For details, see "Prerequisites" on page 63. If you have not yet run these jobs, you can activate them now.
- f** Check the adapter parameters for the HACMP Topology and Application Discovery adapters. To use **sudo** with the commands, adjust the parameters appropriately. They can also be adjusted on the job.

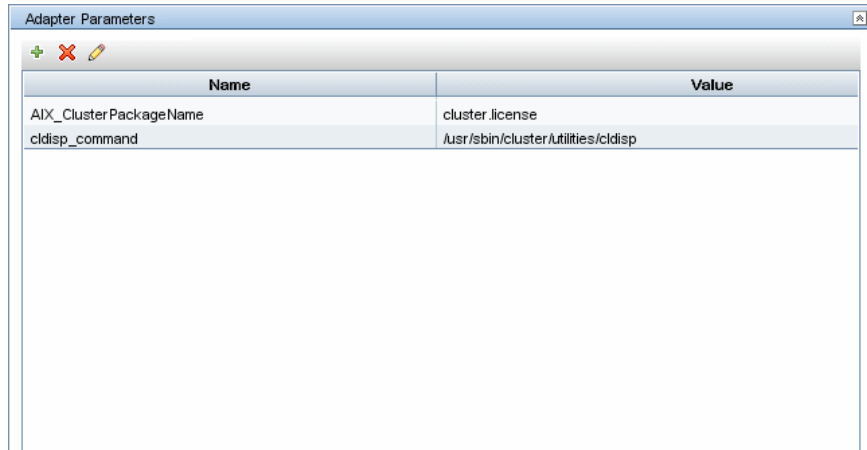
HACMP Application discovery adapters



The screenshot shows a window titled "Adapter Parameters" with a table containing three rows of data. The table has two columns: "Name" and "Value".

Name	Value
cldisp_command	/usr/sbin/cluster/Utilities/cldisp
clsisf_command	clsisf -c
vg_command	lspv

HACMP Topology discovery adapters



- g** Activate the **HACMP Topology Discovery** job, located under the **Cluster – IBM HACMP** module. After the job completes, verify the creation of **HACMP CIs** through the Statistics Results pane. For details, see "Statistics Results Pane" in the *HP Universal CMDB Data Flow Management Guide*.
- h** Activate the **HACMP Application Discovery** job. This job creates HACMP application and resource CIs.

9 Created/Changed Entities

- **HACMP Topology Discovery:**
 - Hacmpcluster CIT
 - Failoverclustersoftware CIT
 - Logical Volume
 - Physical Volume
 - Volume Group
 - Network Interface
- **HACMP Application Discovery:**
 - Hacmpgroup

- Hacmpresource
- Network Interface
- Cluster Server
- IpAddress
- Physical Disk
- Volume Group

Reference

Discovery Mechanism

This section describes the following commands:

- "Verify that the Connected OS Supports HACMP" on page 70
- "Get the Version of HACMP" on page 70
- "Get Cluster Information" on page 71
- "Get DNS Information from the Host File" on page 72
- "Get Volume Group Information" on page 73
- "Get HACMP Application Information" on page 74

Verify that the Connected OS Supports HACMP

Command	uname
Example of output	aix
Values taken	aix
Comments	This command retrieves the OS. This package runs only on AIX platforms so Discovery must verify the OS.

Get the Version of HACMP

Command	lspp -l cluster.license
Example of output	cluster.license 5.4.0.0 COMMITTED HACMP Electronic License
Values taken	5.4.x.x
Comments	This command gives the HACMP version. Discovery verifies that the HACMP version is valid.

Get Cluster Information

Command	/usr/sbin/cluster/utilities/cldisp
Example of output	<pre>## ===== ## Cluster: db590_db591 ## Cluster services: active ## State of cluster: up ## Substate: stable ## ## ##### ## APPLICATIONS ## ##### ## ... ## =====</pre>
Values taken	Cluster: db590_db591
Comments	This command retrieves the HACMP Cluster name.

Get DNS Information from the Host File

Command	cat /etc/hosts
Example of output	<pre>## Sample output... ## ===== ## # Do not remove the following line, or various programs ## # that require network functionality will fail. ## 127.0.0.1 testserver localhost.localdomain localhost ## 12.20.30.3 server1 server1.compay.net ## 12.20.20.3 server1-backup server1- backup.company.net ## 192.168.1.103 server1-local server1- local.company.net ## 12.20.30.4 server2 server1.compay.net ## 12.20.20.4 server2-backup server2- backup.company.net ## 192.168.1.104 server2-local server2- local.company.net ## =====</pre>
Values taken	IP Address and name
Comments	This command retrieves the host name and the IP.

Get Volume Group Information

Command	lspv
Example of output	<pre>## Sample output... # dwscmdb : lspv # hdisk1 00ca4bbe84bdab4f rootvg active # hdisk0 00ca4bbe84bdac14 rootvg active # hdisk2 00ca4bbeeeb6b3c2 QSWIQ9A0_vg concurrent # hdisk3 00ca4bbeeeb3c581 None # hdisk4 00ca4bbeeeb6b499 QSWIQ9A0_vg concurrent # hdisk5 00ca4bbeeeb3c403 None # hdisk6 00ca4bbeeeb6b60d QSWIQ9B0_vg concurrent # hdisk7 00ca4bbeeeb3c4c2 QSWIQ9B0_vg concurrent # hdisk8 00ca4bbeeeb6b84f QSWIQ9A0_vg concurrent # hdisk9 00ca4bbeeeb6b920 QSWIQ9A0_vg concurrent # hdisk10 00ca4bbeeeb3c641 None # hdisk11 00ca4bbeeeb3c7c0 None # hdisk12 00ca4bbeeeb6b6e5 QSWIQ9B0_vg concurrent # hdisk13 00ca4bbeeeb3c700 QSWIQ9B0_vg concurrent</pre>
Values taken	Volume group name
Comments	This command retrieves the volume groups.

Get HACMP Application Information

Command	cldisp
Example of output	<pre> ## Sample output... ## ===== ## Cluster: db590_db591 ## Cluster services: active ## State of cluster: up ## Substate: stable ## ## ##### ## APPLICATIONS ## ##### ## Cluster sy008_sy015 provides the following applications: assy008 ## Application: assy008 {online} ## This application is part of resource group 'ressy008'. ## Resource group policies: ## Startup: on home node only ## Fallover: to next priority node in the list ## Fallback: never ## Nodes configured to provide assy008: a_wwasy008 {up} b_ddasy015 {up} ## Node currently providing assy008: a_wwasy008 {up} ## The node that will provide assy008 if a_wwasy008 fails is: b_ddasy015 ## assy008 is started by /usr/local/bin/start_assy008 ## assy008 is stopped by /usr/local/bin/stop_assy008 </pre>

<p>Example of output (<i>cont'd</i>)</p>	<pre>## Resources associated with assy008: ## Service Labels ## wwasy008(141.122.74.142) {online} ## Interfaces configured to provide wwasy008: ## wwasy008-boot {down} ## with IP address: 141.122.74.149 ## on interface: en1 ## on node: a_wwasy008 {up} ## on network: net_ether_01 {up} ## wwasy008-standby {up} ## with IP address: 192.168.2.40 ## on interface: en2 ## on node: a_wwasy008 {up} ## on network: net_ether_01 {up} ## ddasy015 {up} ## with IP address: 141.122.74.154 ## on interface: en1 ## on node: b_ddasy015 {up} ## on network: net_ether_01 {up} ## ddasy015-standby {up} ## with IP address: 192.168.2.10 ## on interface: en2 ## on node: b_ddasy015 {up} ## on network: net_ether_01 {up} ## Shared Volume Groups: ## vg100 ## vg199 ## No application monitors are configured for assy008.</pre>
---	--

<p>Example of output <i>(cont'd)</i></p>	<pre>## ## ##### ## TOPOLOGY ## ##### ## ... ## =====</pre>
<p>Values taken</p>	<p>Application information</p>
<p>Comments</p>	<p>This command retrieves the HACMP Application information.</p>

5

Microsoft Cluster

This chapter includes:

Tasks

- ▶ Discover Microsoft Cluster Servers on page 78

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 78
- "Discovery Workflow" on page 78
- "Topology Map" on page 79

1 Network and Protocols

For credentials information, see:

- "WMI Protocol"
- "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:

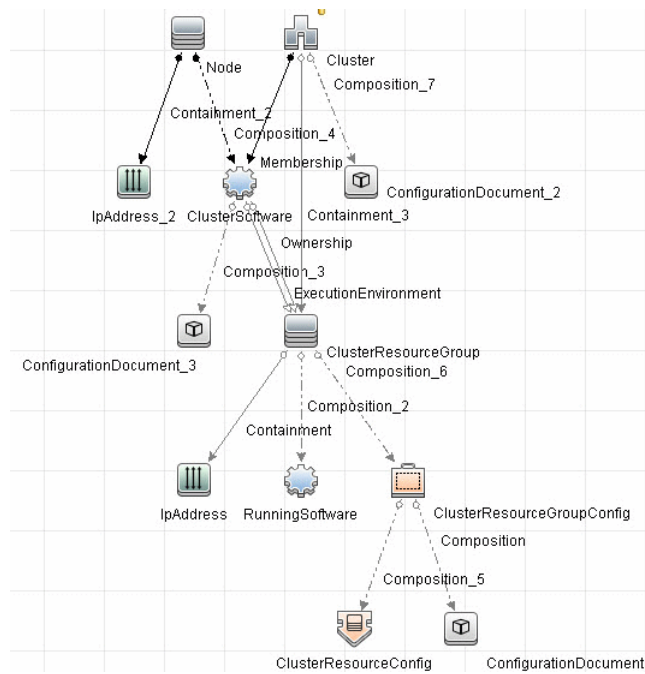
- **Network – Basic** (Host Connection by Shell)
- **Network – Host Resources and Applications**
- **Cluster – Microsoft Cluster** (MS Cluster by NTCMD)

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.



6

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 82
- ▶ Discovery Mechanism on page 82
- ▶ Discovering NLB with the Command Line Utility on page 83

Tasks

- ▶ Discover Microsoft Network Load Balancing Systems on page 86

Reference

- ▶ MS NLB Cluster CIT on page 91
- ▶ NLB Cluster Software CIT on page 92
- ▶ ConfigurationDocument (NLB Port Rule) on page 93
- ▶ Glossary on page 94
- ▶ Components of the Network Load Balancing Architecture on page 95

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 83.

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 86
- "Discovery Workflow" on page 87
- "Packages" on page 87
- "Discovered CITs" on page 88
- "Trigger Query" on page 89
- "MS NLB by NTCMD Adapter" on page 89
- "Views" on page 90
- "Topology" on page 90

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 82.

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

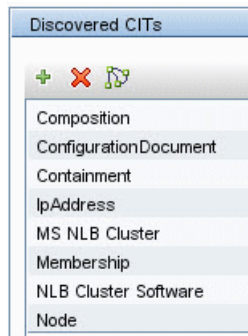
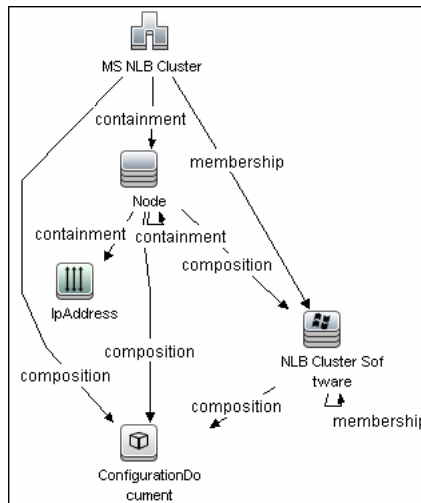
3 Packages

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

For details, see "Package Manager" in the *HP Universal CMDB Administration Guide*.

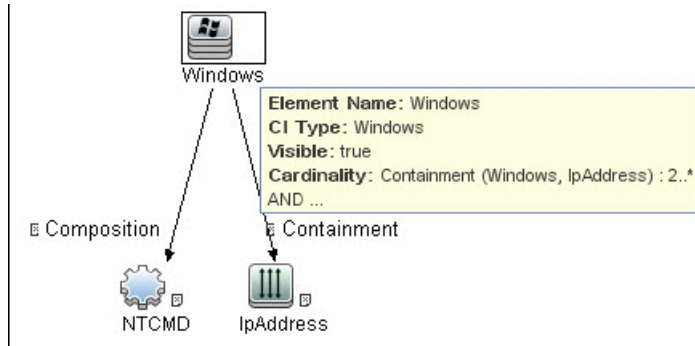
4 Discovered CITs

DFM discovers the following CITs:



- ▶ MS NLB Cluster. For details, see "MS NLB Cluster CIT" on page 91.
- ▶ NLB Cluster Software. For details, see "NLB Cluster Software CIT" on page 92.
- ▶ Configuration File. For details, see "ConfigurationDocument (NLB Port Rule)" on page 93.

5 Trigger Query



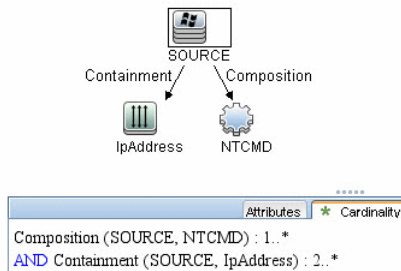
Condition. NTCMD running on a Windows machine with at least two IP addresses.

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



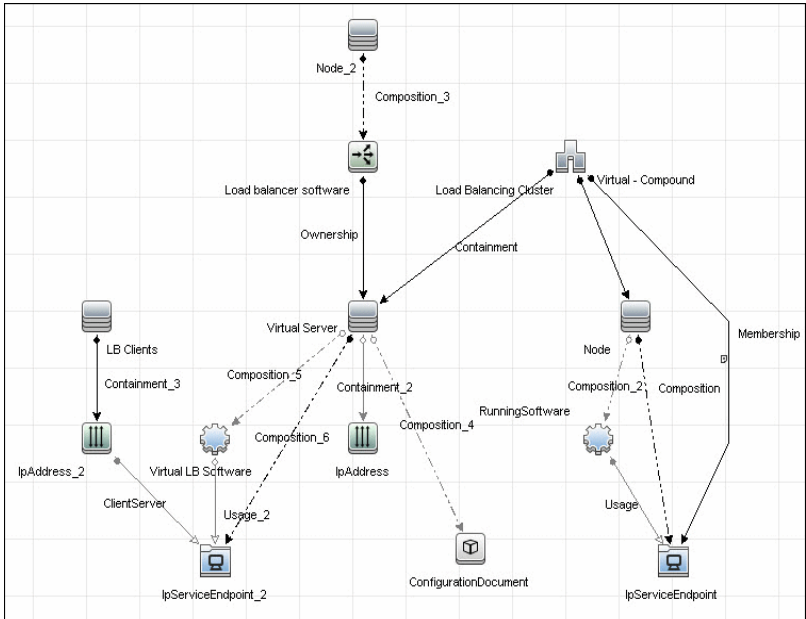
Triggered CI Data.

Name	Value
credentialsId	\${NTCMD.credentials_id}
ip_address	\${IpAddress.name}

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..*	membership	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

Key	Display Name	Attribute Name	Type
	MulticastSupportEnable	multicast_support_enable	Boolean
	IGMPSupport	igmp_support	Boolean
	RemoteControlEnabled	remote_control_enabled	Boolean
X	Name	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

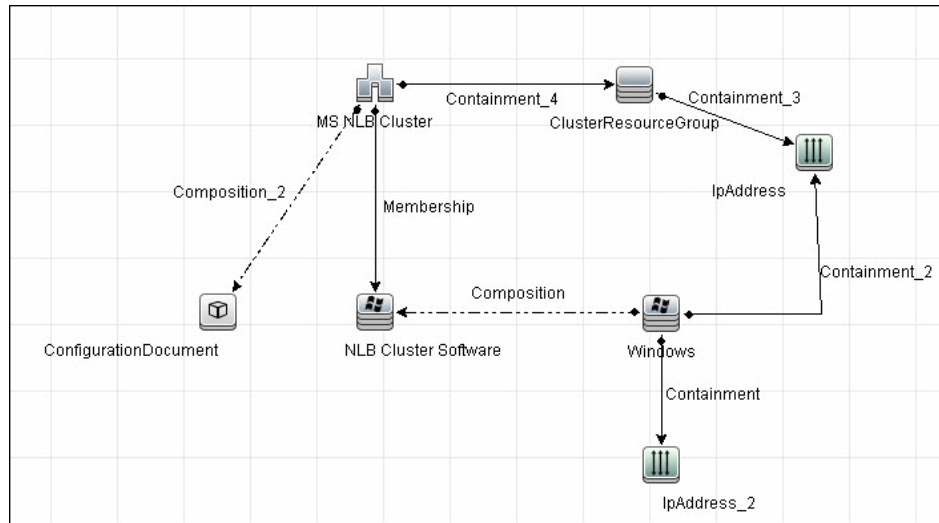
Links

Start Node	Start Node Cardinality	Link Name	End Node	End Node Cardinality
ms_nlb_cluster	1..*	membership	nlb_clustersoftware	1..*
nt	1..*	composition	nlb_clustersoftware	1..*

Attributes

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

ConfigurationDocument (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..*	composition	nlb_clustersoftware	1..*
ms_nlb_cluster	1..*	membership	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.

7

Sun Cluster

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 98

Tasks

- ▶ Discover Sun Cluster on page 99

Reference

- ▶ Discovery Mechanism on page 104

Concepts

Overview

The Sun Cluster product is an integrated hardware and software solution used to create highly available and scalable services. The Sun Cluster environment extends the Solaris Operating System into a cluster operating system. A cluster is a collection of one or more nodes that belong exclusively to that collection.

Tasks

Discover Sun Cluster

This task includes the following steps:

- "Supported Version" on page 99
- "Prerequisites" on page 99
- "Permissions" on page 100
- "Trigger Query for the Solaris Cluster by Shell Job" on page 100
- "The Input Query" on page 101
- "Discovery Workflow" on page 101
- "Sample Output" on page 102
- "Created/Changed Entities" on page 102

1 Supported Version

The **Sun Cluster** package supports Sun Cluster 3.2. Support for older versions of Sun Cluster has not been verified.

The Sun Cluster software integrates with the Solaris operating system, thus only this OS is supported.

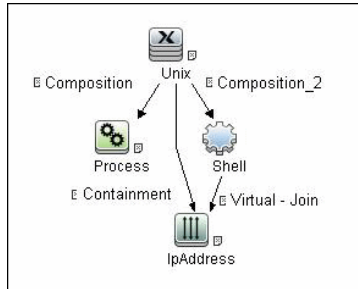
2 Prerequisites

- Configure the appropriate credentials:
 - For credentials information, see "SSH Protocol" and "Telnet Protocol" in *HP Universal CMDB Data Flow Management Guide*.

3 Permissions

Set up permissions for users performing Sun Cluster discovery to run clustering commands (`scrgadm`, `scstat`, `scconf`, and so on). For a full list of commands see "Discovery Mechanism" on page 104.

4 Trigger Query for the Solaris Cluster by Shell Job



Process Attributes:

Element name: Visible Include subtypes

Attribute Cardinality Qualifier Identity

Advanced layout settings

NOT	(Criteria)	And/Or
<input type="checkbox"/>		Name Equal ignore case "cluster"		

Shell Attributes:

Element name: Visible Include subtypes

Attribute Cardinality Qualifier Identity

Advanced layout settings

NOT	(Criteria)	And/Or
<input checked="" type="checkbox"/>		Reference to the credentials dictionary entry Is null		

IpAddress Attributes:

Element name: Visible Include subtypes

Attribute Cardinality Qualifier Identity

Advanced layout settings

NOT	(Criteria)	And/Or
<input checked="" type="checkbox"/>		IP Probe Name Is null		

5 The Input Query

This query contains only one Shell CI:

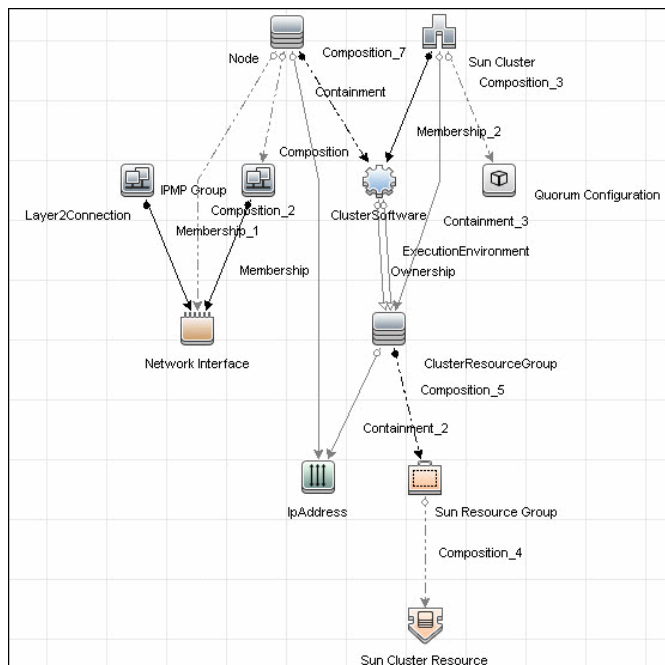


6 Discovery Workflow

For details on jobs, see "Discovery Control Panel – Advanced Mode Workflow" in *HP Universal CMDB Data Flow Management Guide*.

- a** Run the **Range IPs by ICMP** job to discover which of the machines in the IP range are up.
- b** Run the **Host Connection by Shell** job to discover Shell connectivity and basic information about the hosts.
- c** Run the **Host Resources and Applications by Shell** job to discover processes on the target machines.
- d** Run the **Sun Cluster by Shell** job to discover the Sun Cluster topology.

7 Sample Output



8 Created/Changed Entities

- Added CI Types:
 - Sun Cluster
 - Sun Resource Group
 - Sun Cluster Resource
 - IPMP Group
- Added valid links:
 - Node - composition > IPMP Group
 - IPMP Group - membership > Network Interface
- Added views:
 - Sun Cluster Topology view

- Added scripts:
 - sun_cluster_by_shell.py
 - solaris_networking.py
- Added adapters:
 - Sun_Cluster_by_Shell
- Added jobs:
 - Sun Cluster by Shell
- Added Trigger query:
 - shell_on_solaris_cs
- Added module:
 - Sun Cluster.xml

Reference

Discovery Mechanism

This section includes the Sun clustering commands:

- "Get Name of Cluster" on page 105
- "Get Nodes of Cluster" on page 106
- "Resolve Node Names to IPs" on page 106
- "Get Status of Nodes" on page 107
- "Get Resource Groups and Resources" on page 107
- "Get Details for Resource Groups and Resources" on page 109
- "Get Cluster Interconnection Information" on page 122
- "Get Quorum Configuration" on page 126

Get Name of Cluster

Command	/usr/cluster/bin/scconf -p
Example of output	<pre> Cluster name: cluster1 Cluster ID: 0x4A7BD3D3 Cluster install mode: disabled Cluster private net: 172.2.0.0 Cluster private netmask: 255.255.255.192 Cluster maximum nodes: 6 Cluster maximum private networks: 4 Cluster new node authentication: unix Cluster authorized-node list: <. - Exclude all nodes> Cluster transport heart beat timeout: 10000 Cluster transport heart beat quantum: 1000 Round Robin Load Balancing UDP session timeout: 480 Cluster nodes: node1 node2 Cluster node name: node1 ... </pre>
Values taken	Name of the cluster: cluster1
Comments	Name of the cluster enabling the creation of the Sun Cluster CI.

Get Nodes of Cluster

Command	<code>/usr/cluster/bin/scconf -p</code>
Example of output	<pre>Cluster name: cluster1 Cluster ID: 0x4A7BDCD3 Cluster install mode: disabled Cluster private net: 172.2.7.0 Cluster private netmask: 255.255.255.192 Cluster maximum nodes: 6 Cluster maximum private networks: 4 Cluster new node authentication: unix Cluster authorized-node list: <. - Exclude all nodes> Cluster transport heart beat timeout: 10000 Cluster transport heart beat quantum: 1000 Round Robin Load Balancing UDP session timeout: 480 Cluster nodes: node1 node2 ...</pre>
Values taken	Node names

Resolve Node Names to IPs

Command	<code>/usr/sbin/nslookup node1</code>
Example of output	<pre>Server: 134.44.0.10 Address: 134.44.0.10#53 Name: node1.example.com Address: 134.44.0.75</pre>

Values taken	IP of the node.
Comments	The IP enables the creation of an incomplete Host for each node in the cluster.

Get Status of Nodes

Command	/usr/cluster/bin/scstat -n
Example of output	<pre>-- Cluster Nodes -- Node name Status ----- -</pre> <p>Cluster node: node1 Online</p> <p>Cluster node: node2 Online</p>
Values taken	Node statuses
Comments	Although statuses are not reported, Discovery needs this status. For example, Discovery should not issue an arp command to resolve the MAC address if the node is off-line.

Get Resource Groups and Resources

Command	/usr/cluster/bin/scstat -g
----------------	----------------------------

<p>Example of output</p>	<pre>-- Resource Groups and Resources -- Group Name Resources ----- - Resources: oracle1 oracle1-zfs oracle1-lh oracle1- ora oracle1-cron oracle1-lsnr_ano_10 -- Resource Groups -- ...</pre>
<p>Values taken</p>	<p>List of groups. List of resources in a group. Status of a group on each of the nodes (run links are created based on this).</p>

Get Details for Resource Groups and Resources

Command	/usr/cluster/bin/scrgadm -pvv
Example of output	<pre> Res Group name: oracle1 (oracle1) Res Group RG_description: <NULL> (oracle1) Res Group mode: Failover (oracle1) Res Group management state: Managed (oracle1) Res Group RG_project_name: user.oracle (oracle1) Res Group RG_SLM_type: manual (oracle1) Res Group RG_affinities: <NULL> (oracle1) Res Group Auto_start_on_new_cluster: True (oracle1) Res Group Failback: False (oracle1) Res Group Nodelist: node1 node2 (oracle1) Res Group Maximum primaries: 1 (oracle1) Res Group Desired primaries: 1 (oracle1) Res Group RG_dependencies: <NULL> (oracle1) Res Group network dependencies: True (oracle1) Res Group Global_resources_used: <All> (oracle1) Res Group Pingpong_interval: 3600 (oracle1) Res Group Pathprefix: <NULL> (oracle1) Res Group system: False (oracle1) Res Group Suspend_automatic_recovery: False (oracle1) Res name: oracle1-zfs (oracle1:oracle1-zfs) Res R_description: (oracle1:oracle1-zfs) Res resource type: SUNW.HAStoragePlus:8 </pre>

<p>Example of output (<i>cont'd</i>)</p>	<pre>(oracle1:oracle1-zfs) Res type version: 8 (oracle1:oracle1-zfs) Res resource group name: oracle1 (oracle1:oracle1-zfs) Res resource project name: user.oracle (oracle1:oracle1-zfs{kvsdb1}) Res enabled: True (oracle1:oracle1-zfs{kvsdb2}) Res enabled: True (oracle1:oracle1-zfs{kvsdb1}) Res monitor enabled: True (oracle1:oracle1-zfs{kvsdb2}) Res monitor enabled: True (oracle1:oracle1-zfs) Res strong dependencies: <NULL> (oracle1:oracle1-zfs) Res weak dependencies: <NULL> (oracle1:oracle1-zfs) Res restart dependencies: <NULL> (oracle1:oracle1-zfs) Res offline restart dependencies: <NULL> (oracle1:oracle1-zfs) Res property name: Retry_interval (oracle1:oracle1-zfs:Retry_interval) Res property class: standard (oracle1:oracle1-zfs:Retry_interval) Res property description: Time in which monitor attempts to restart a failed resource Retry_count times. (oracle1:oracle1-zfs:Retry_interval) Res property type: int (oracle1:oracle1-zfs:Retry_interval) Res property value: 300 (oracle1:oracle1-zfs) Res property name: Retry_count (oracle1:oracle1-zfs:Retry_count) Res property class: standard (oracle1:oracle1-zfs:Retry_count) Res property description: Indicates the number of times a monitor restarts the resource if it fails. (oracle1:oracle1-zfs:Retry_count) Res property type: int (oracle1:oracle1-zfs:Retry_count) Res property value: 2 (oracle1:oracle1-zfs) Res property name: Failover_mode (oracle1:oracle1-zfs:Failover_mode) Res property class: standard</pre>
---	---

<p>Example of output (cont'd)</p>	<p>(oracle1:oracle1-zfs:Failover_mode) Res property description: Modifies recovery actions taken when the resource fails.</p> <p>(oracle1:oracle1-zfs:Failover_mode) Res property type: enum</p> <p>(oracle1:oracle1-zfs:Failover_mode) Res property value: SOFT</p> <p>(oracle1:oracle1-zfs) Res property name: POSTNET_STOP_TIMEOUT</p> <p>(oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property description: Maximum execution time allowed for Postnet_stop method.</p> <p>(oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property value: 1800</p> <p>(oracle1:oracle1-zfs) Res property name: PRENET_START_TIMEOUT</p> <p>(oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property description: Maximum execution time allowed for Prenet_Start method.</p> <p>(oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property value: 1800</p> <p>(oracle1:oracle1-zfs) Res property name: MONITOR_CHECK_TIMEOUT</p> <p>(oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property description: Maximum execution time allowed for Monitor_Check method.</p>
--	--

<p>Example of output (<i>cont'd</i>)</p>	<p>(oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property value: 90</p> <p>(oracle1:oracle1-zfs) Res property name: MONITOR_STOP_TIMEOUT</p> <p>(oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property description: Maximum execution time allowed for Monitor_Stop method.</p> <p>(oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property type: int (oracle1:oracle1- zfs:MONITOR_STOP_TIMEOUT) Res property value: 90</p> <p>(oracle1:oracle1-zfs) Res property name: MONITOR_START_TIMEOUT</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property description: Maximum execution time allowed for Monitor_Start method.</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property value: 90</p> <p>(oracle1:oracle1-zfs) Res property name: INIT_TIMEOUT</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property description: Maximum execution time allowed for Init method.</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property value: 1800</p> <p>(oracle1:oracle1-zfs) Res property name: UPDATE_TIMEOUT</p>
---	--

<p>Example of output (cont'd)</p>	<p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property description: Maximum execution time allowed for Update method.</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property value: 1800</p> <p>(oracle1:oracle1-zfs) Res property name: VALIDATE_TIMEOUT</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property description: Maximum execution time allowed for Validate method.</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property value: 1800</p> <p>(oracle1:oracle1-zfs) Res property name: ZpoolsSearchDir</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property class: extension</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property description: Directory location to search devices for zpools</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property pernode: False</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property type: string</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property value:</p> <p>(oracle1:oracle1-zfs) Res property name: FilesystemCheckCommand</p> <p>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property class: extension</p>
--	--

<p>Example of output (<i>cont'd</i>)</p>	<pre>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property description: Command string to be executed for file system checks (oracle1:oracle1-zfs:FilesystemCheckCommand) Res property pernode: False (oracle1:oracle1-zfs:FilesystemCheckCommand) Res property type: stringarray (oracle1:oracle1-zfs:FilesystemCheckCommand) Res property value: <NULL> (oracle1:oracle1-zfs) Res property name: AffinityOn (oracle1:oracle1-zfs:AffinityOn) Res property class: extension (oracle1:oracle1-zfs:AffinityOn) Res property description: For specifying affinity switchover (oracle1:oracle1-zfs:AffinityOn) Res property pernode: False (oracle1:oracle1-zfs:AffinityOn) Res property type: boolean (oracle1:oracle1-zfs:AffinityOn) Res property value: TRUE (oracle1:oracle1-zfs) Res property name: FilesystemMountPoints (oracle1:oracle1-zfs:FilesystemMountPoints) Res property class: extension (oracle1:oracle1-zfs:FilesystemMountPoints) Res property description: The list of file system mountpoints (oracle1:oracle1-zfs:FilesystemMountPoints) Res property pernode: False (oracle1:oracle1-zfs:FilesystemMountPoints) Res property type: stringarray (oracle1:oracle1-zfs:FilesystemMountPoints) Res property value: <NULL> (oracle1:oracle1-zfs) Res property name: GlobalDevicePaths (oracle1:oracle1-zfs:GlobalDevicePaths) Res property class: extension (oracle1:oracle1-zfs:GlobalDevicePaths) Res property description: The list of HA global device paths</pre>
---	---

<p>Example of output (cont'd)</p>	<pre>(oracle1:oracle1-zfs:GlobalDevicePaths) Res property pernode: False (oracle1:oracle1-zfs:GlobalDevicePaths) Res property type: stringarray (oracle1:oracle1-zfs:GlobalDevicePaths) Res property value: <NULL> (oracle1:oracle1-zfs) Res property name: Zpools (oracle1:oracle1-zfs:Zpools) Res property class: extension (oracle1:oracle1-zfs:Zpools) Res property description: The list of zpools (oracle1:oracle1-zfs:Zpools) Res property pernode: False (oracle1:oracle1-zfs:Zpools) Res property type: stringarray (oracle1:oracle1-zfs:Zpools) Res property value: oracle1prod (oracle1) Res name: oracle1-lh (oracle1:oracle1-lh) Res R_description: (oracle1:oracle1-lh) Res resource type: SUNW.LogicalHostname:2 (oracle1:oracle1-lh) Res type version: 2 (oracle1:oracle1-lh) Res resource group name: oracle1 (oracle1:oracle1-lh) Res resource project name: user.oracle (oracle1:oracle1-lh{kvsdb1}) Res enabled: True (oracle1:oracle1-lh{kvsdb2}) Res enabled: True (oracle1:oracle1-lh{kvsdb1}) Res monitor enabled: True (oracle1:oracle1-lh{kvsdb2}) Res monitor enabled: True (oracle1:oracle1-lh) Res strong dependencies: <NULL> (oracle1:oracle1-lh) Res weak dependencies: <NULL> (oracle1:oracle1-lh) Res restart dependencies: <NULL> (oracle1:oracle1-lh) Res offline restart dependencies: <NULL> (oracle1:oracle1-lh) Res property name: Retry_interval</pre>
--	--

<p>Example of output (cont'd)</p>	<pre>(oracle1:oracle1-lh:Retry_interval) Res property class: standard (oracle1:oracle1-lh:Retry_interval) Res property description: Time in which monitor attempts to restart a failed resource Retry_count times. (oracle1:oracle1-lh:Retry_interval) Res property type: int (oracle1:oracle1-lh:Retry_interval) Res property value: 300 (oracle1:oracle1-lh) Res property name: Retry_count (oracle1:oracle1-lh:Retry_count) Res property class: standard (oracle1:oracle1-lh:Retry_count) Res property description: Indicates the number of times a monitor restarts the resource if it fails. (oracle1:oracle1-lh:Retry_count) Res property type: int (oracle1:oracle1-lh:Retry_count) Res property value: 2 (oracle1:oracle1-lh) Res property name: Thorough_probe_interval (oracle1:oracle1-lh:Thorough_probe_interval) Res property class: standard (oracle1:oracle1-lh:Thorough_probe_interval) Res property description: Time between invocations of a high-overhead fault probe of the resource. (oracle1:oracle1-lh:Thorough_probe_interval) Res property type: int (oracle1:oracle1-lh:Thorough_probe_interval) Res property value: 60 (oracle1:oracle1-lh) Res property name: Cheap_probe_interval (oracle1:oracle1-lh:Cheap_probe_interval) Res property class: standard (oracle1:oracle1-lh:Cheap_probe_interval) Res property description: Time between invocations of a quick fault probe of the resource. (oracle1:oracle1-lh:Cheap_probe_interval) Res property type: int</pre>
--	---

<p>Example of output (cont'd)</p>	<pre>(oracle1:oracle1-lh:Cheap_probe_interval) Res property value: 60 (oracle1:oracle1-lh) Res property name: Failover_mode (oracle1:oracle1-lh:Failover_mode) Res property class: standard (oracle1:oracle1-lh:Failover_mode) Res property description: Modifies recovery actions taken when the resource fails. (oracle1:oracle1-lh:Failover_mode) Res property type: enum (oracle1:oracle1-lh:Failover_mode) Res property value: HARD (oracle1:oracle1-lh) Res property name: PRENET_START_TIMEOUT (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property description: Maximum execution time allowed for Prenet_Start method. (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property type: int (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property value: 300 (oracle1:oracle1-lh) Res property name: MONITOR_CHECK_TIMEOUT (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property description: Maximum execution time allowed for Monitor_Check method. (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property type: int (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property value: 300</pre>
--	---

<p>Example of output (<i>cont'd</i>)</p>	<pre> (oracle1:oracle1-lh) Res property name: MONITOR_STOP_TIMEOUT (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property description: Maximum execution time allowed for Monitor_Stop method. (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property type: int (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property value: 300 (oracle1:oracle1-lh) Res property name: MONITOR_START_TIMEOUT (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property description: Maximum execution time allowed for Monitor_Start method. (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property type: int (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property value: 300 (oracle1:oracle1-lh) Res property name: UPDATE_TIMEOUT (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property description: Maximum execution time allowed for Update method. (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property type: int (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property value: 300 (oracle1:oracle1-lh) Res property name: VALIDATE_TIMEOUT </pre>
---	---

<p>Example of output (cont'd)</p>	<p>(oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property description: Maximum execution time allowed for Validate method.</p> <p>(oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property value: 300</p> <p>(oracle1:oracle1-lh) Res property name: STOP_TIMEOUT</p> <p>(oracle1:oracle1-lh:STOP_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-lh:STOP_TIMEOUT) Res property description: Maximum execution time allowed for Stop method.</p> <p>(oracle1:oracle1-lh:STOP_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-lh:STOP_TIMEOUT) Res property value: 300</p> <p>(oracle1:oracle1-lh) Res property name: START_TIMEOUT</p> <p>(oracle1:oracle1-lh:START_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-lh:START_TIMEOUT) Res property description: Maximum execution time allowed for Start method.</p> <p>(oracle1:oracle1-lh:START_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-lh:START_TIMEOUT) Res property value: 500</p> <p>(oracle1:oracle1-lh) Res property name: CheckNameService</p> <p>(oracle1:oracle1-lh:CheckNameService) Res property class: extension</p> <p>(oracle1:oracle1-lh:CheckNameService) Res property description: Name service check flag</p> <p>(oracle1:oracle1-lh:CheckNameService) Res property pernode: False</p>
--	--

<p>Example of output (<i>cont'd</i>)</p>	<pre>(oracle1:oracle1-lh:CheckNameService) Res property type: boolean (oracle1:oracle1-lh:CheckNameService) Res property value: TRUE (oracle1:oracle1-lh) Res property name: NetIfList (oracle1:oracle1-lh:NetIfList) Res property class: extension (oracle1:oracle1-lh:NetIfList) Res property description: List of IPMP groups on each node (oracle1:oracle1-lh:NetIfList) Res property pernode: False (oracle1:oracle1-lh:NetIfList) Res property type: stringarray (oracle1:oracle1-lh:NetIfList) Res property value: ipmp1@1 ipmp1@2 (oracle1:oracle1-lh) Res property name: HostnameList (oracle1:oracle1-lh:HostnameList) Res property class: extension (oracle1:oracle1-lh:HostnameList) Res property description: List of hostnames this resource manages (oracle1:oracle1-lh:HostnameList) Res property pernode: False (oracle1:oracle1-lh:HostnameList) Res property type: stringarray (oracle1:oracle1-lh:HostnameList) Res property value: oracle1 ... </pre>
---	---

Values taken	<ul style="list-style-type: none"> ➤ Groups: <ul style="list-style-type: none"> ➤ Name ➤ Description ➤ Management state ➤ Mode (failover/scalable) ➤ Maximum primaries ➤ Desired primaries ➤ Nodes list ➤ Is system ➤ Autostart on new cluster ➤ Failback ➤ Resources: <ul style="list-style-type: none"> ➤ Name ➤ Description ➤ Type ➤ Failover mode ➤ Retry interval ➤ Retry count
Comments	<p>Based on the extracted value, Discovery creates Resource Groups with attributes and Resources with attributes.</p> <p>LogicalHostname handling: for this type of resource Discovery extracts an additional HostnameList property that contains the host names that this resource manages. Host names are resolved to IPs. Resolved IPs are attached to the ClusteredServer CIT.</p>

Get Cluster Interconnection Information

Command	<code>/usr/cluster/bin/scstat -W</code>
Example of output	<pre>-- Cluster Transport Paths -- Endpoint Endpoint Status ----- -</pre> <p>Transport path: node1:bge3 node2:nxge11 Path online</p> <p>Transport path: node1:nxge3 node2:nxge3 Path online</p>
Values taken	<p>Output contains the list of transport paths with their statuses.</p> <p>For each path which is online we get source interface on a source node and target interface on a target node.</p>
Comments	<p>Such transport path will be reported with Layer2 links from source interface to target interface.</p> <p>To report the remote interface (located on a node which is not the one connected to), the MAC addresses described below are retrieved.</p>

Command	/usr/cluster/bin/scconf -p
Example of output	<pre> ... Cluster install mode: disabled Cluster private net: 172.2.0.0 Cluster private netmask: 255.255.255.192 Cluster maximum nodes: 6 Cluster maximum private networks: 4 Cluster new node authentication: unix Cluster authorized-node list: <. - Exclude all nodes> Cluster transport heart beat timeout: 10000 Cluster transport heart beat quantum: 1000 Round Robin Load Balancing UDP session timeout: 480 Cluster nodes: node1 node2 Cluster node name: node1 Node ID: 1 Node enabled: yes Node private hostname: clusternode1-priv Node quorum vote count: 1 Node reservation key: 0x4A7ADDD300000001 Node zones: <NULL> CPU shares for global zone: 1 Minimum CPU requested for global zone: 1 Node transport adapters: nxge3 bge3 Node transport adapter: nxge3 Adapter enabled: yes Adapter transport type: dlpi Adapter property: device_name=nxge </pre>

<p>Example of output (<i>cont'd</i>)</p>	<pre> Adapter property: device_instance=3 Adapter property: lazy_free=1 Adapter property: dlpi_heartbeat_timeout=10000 Adapter property: dlpi_heartbeat_quantum=1000 Adapter property: nw_bandwidth=80 Adapter property: bandwidth=70 Adapter property: ip_address=172.2.0.9 Adapter property: netmask=255.255.255.248 Adapter port names: 0 Adapter port: 0 Port enabled: yes Node transport adapter: bge3 Adapter enabled: yes Adapter transport type: dlpi Adapter property: device_name=bge Adapter property: device_instance=3 Adapter property: lazy_free=1 Adapter property: dlpi_heartbeat_timeout=10000 Adapter property: dlpi_heartbeat_quantum=1000 Adapter property: nw_bandwidth=80 Adapter property: bandwidth=70 Adapter property: ip_address=172.2.0.17 Adapter property: netmask=255.255.255.248 Adapter port names: 0 Adapter port: 0 Port enabled: yes ... </pre>
---	---

Values taken	Private network address. List of interfaces that are used in cluster interconnect: name and IP address assigned.
---------------------	--

Command	/usr/sbin/arp 172.2.0.10
Example of output	172.2.0.10 (172.2.0.10) at 0:21:a8:39:33:a9
Values taken	MAC
Comments	Discovery resolves the MAC address of remote interface via arp. If it cannot be resolved, Discovery does not report the transport path as Layer2 link.

Get Quorum Configuration

Command	<code>/usr/cluster/bin/scstat -q</code>
Example of output	<pre>-- Quorum Summary from latest node reconfiguration -- Quorum votes possible: 3 Quorum votes needed: 2 Quorum votes present: 3 -- Quorum Votes by Node (current status) -- Node Name Present Possible Status ----- - Node votes: node1 1 1 Online Node votes: node2 1 1 Online -- Quorum Votes by Device (current status) -- Device Name Present Possible Status ----- - Device votes: clusterquo1 1 1 Online</pre>
Values taken	The quorum status information.
Comments	The details about quorum devices are appended to the Quorum Configuration config file.

8

Veritas

This chapter includes:

Tasks

- ▶ Discover Veritas Cluster Servers on page 128

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 128
- "Network and Protocols" on page 128
- "Discovery Workflow" on page 129
- "Discovered CITs" on page 129
- "Topology Map" on page 130

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

2 Network and Protocols

For credentials information, see:

- "SSH Protocol"
- "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

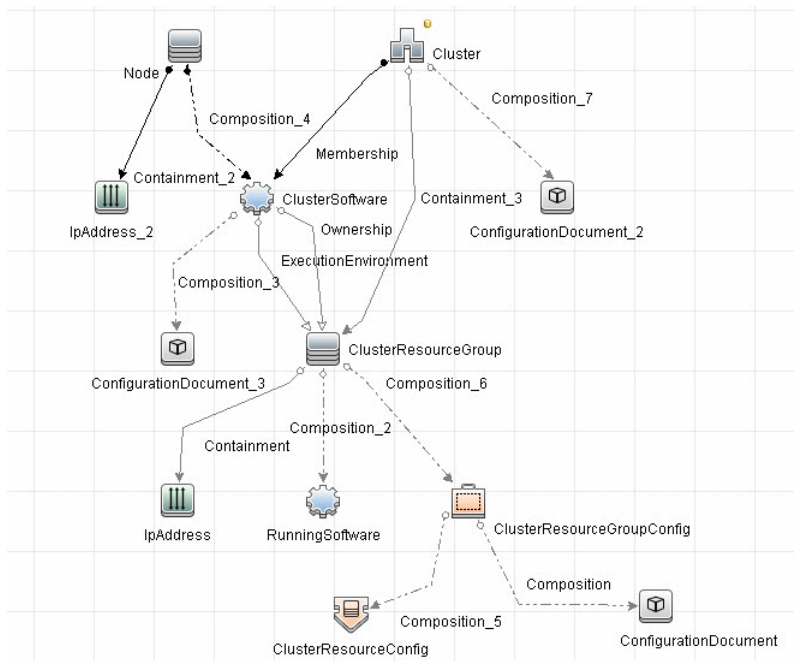
To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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 **membership** relationship to the Veritas Cluster.



9

Database Connections by Host Credentials

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- Overview on page 132
- Discovery Mechanism on page 132

Tasks

- Discover Database Connections by Host Credentials on page 134

Concepts

Overview

The purpose of this package is to enable database auto-discovery using host level credentials in HP Universal CMDB (UCMDB). In certain cases, a DFM user or administrator does not have detailed information about the database, such as its name or SID, listener port number, and so on. The solution in this package discovers this information with minimal inputs, and enables end-to-end discovery of databases.

DFM extracts database information from various sources, for example, from running process names, Windows service names, the Windows registry, and configuration files, on the database server and build CIs. Discovered Database CIs can be used as triggers for the Database Connection by SQL jobs (for example, the **Oracle Database Connection by SQL** job), to populate database credentials, thus enabling deep discovery using out-of-the-box database topology discovery jobs.

Discovery Mechanism

DFM triggers for jobs in this package are set up so that these jobs are seamlessly included in the UCMDB spiral discovery schedule.

The **DB Connections by Shell** and **DB Connections by WMI** jobs in this package use a Shell (NTCMD/SSH/Telnet) or agent (WMI) CI as a trigger, to search for database signatures on a host. These jobs create database CIs with available information, such as instance name or SID and the listener port of the database server. Since database credentials are not used, the username and credentials ID attributes of these CIs are empty.

This section also includes:

- ▶ "DB Connections by Shell Job" on page 133
- ▶ "DB Connections by WMI Job" on page 133

DB Connections by Shell Job

This discovery job attempts to identify configured databases on a host using a Shell client (NTCMD/SSH/Telnet). Once connected, the job creates a list of running processes and server ports associated with each process. On Microsoft Windows operating systems, this job adds a list of installed Windows services to the list.

The job then looks for known database signatures in this list of processes and services, to create database CIs.

Mapping ports to processes can require specific privileges depending on the operating system in use. If the necessary privileges are not available, this job attempts to create database CIs using the available information. However, details may be missing, for example, the database port. In such cases, you may need to run the job again after entering new credentials with the necessary privileges. For details on adding credentials, see "Domain Credential References" in *HP Universal CMDB Data Flow Management Guide*.

After identifying databases using the above information, this job attempts to retrieve additional information on configured (but not running) instances from registry keys (on Microsoft Windows only) and by parsing well known configuration files.

DB Connections by WMI Job

Similarly to the **DB Connections by Shell** job, this job attempts to create a list of processes and services, and parses them for database signatures.

Since an agent does not have access to output of commands such as **netstat**, this job is limited in that the listener ports of database servers are not always identified. Port information for databases such as Microsoft SQL Server is available in the Windows registry, and this job queries that information when connected through WMI.

Tasks

Discover Database Connections by Host Credentials

This task includes the following steps:

- "Supported Versions" on page 134
- "Set up Protocols" on page 135
- "Discovery Workflow" on page 135
- "Trigger Query for the DB Connection by Shell Job" on page 136
- "Input Query for the DB Connection by Shell Job" on page 137
- "Trigger Query for the DB Connection by WMI Job" on page 137
- "Input Query for the DB Connection by WMI Job" on page 137
- "Adapter Parameters for the DB Connections by Shell job" on page 138
- "Adapter Parameters for the DB Connections by WMI job" on page 139
- "Discovered CITs" on page 140
- "Sample Output" on page 140

1 Supported Versions

This discovery solution supports the following database servers:

- Oracle 9i, 10g, 11g
- Microsoft SQL Server 2000, 2005, 2008
- IBM DB2 8.x and 9.x

2 Set up Protocols

For credentials information, see:

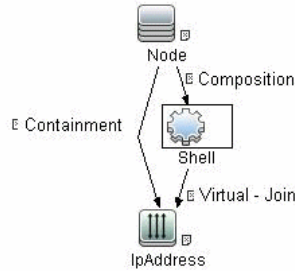
- "WMI Protocol"
- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

3 Discovery Workflow

- a** Run the **Range IPs by ICMP** job (**Discovery Modules > Network Discovery > Basic**).
- b** Run the **Host Connection by Shell** job (**Discovery Modules > Network Discovery > Basic**).
- c** Run the **Host Connection by WMI** job (**Discovery Modules > Network Discovery > Basic**).
- d** Run the **DB Connections by Shell** job (**Discovery Modules > Database – Connections Using Host Credentials**). For details, see "DB Connections by Shell Job" on page 133.
- e** Run the **DB Connections by WMI** job (**Discovery Modules > Database – Connections Using Host Credentials**). For details, see "DB Connections by WMI Job" on page 133.

4 Trigger Query for the DB Connection by Shell Job



* Attributes * Cardinality Qualifiers Selected Identities * Details

NOT Reference to the credentials dictionary entry Is null

Shell Attributes:

Element name: Shell Visible Include subtypes

Attribute Cardinality Qualifier Identity

Advanced layout settings

NOT	(Criteria)	And/Or
<input checked="" type="checkbox"/>		Reference to the credentials dictionary entry Is null		

IPAddress Attributes:

Query Node Properties

Query Node Properties
Enables you to add attributes, cardinality, qualifiers and CI specific conditions

Element name: IPAddress Visible Include subtypes

Attribute Cardinality Qualifier Identity

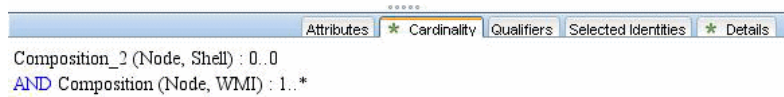
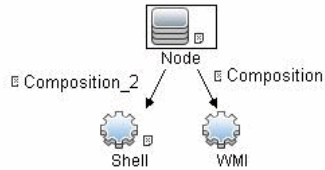
Advanced layout settings

NOT	(Criteria)	And/Or
<input checked="" type="checkbox"/>		IP Probe Name Is null		

5 Input Query for the DB Connection by Shell Job

There is no Input Query.

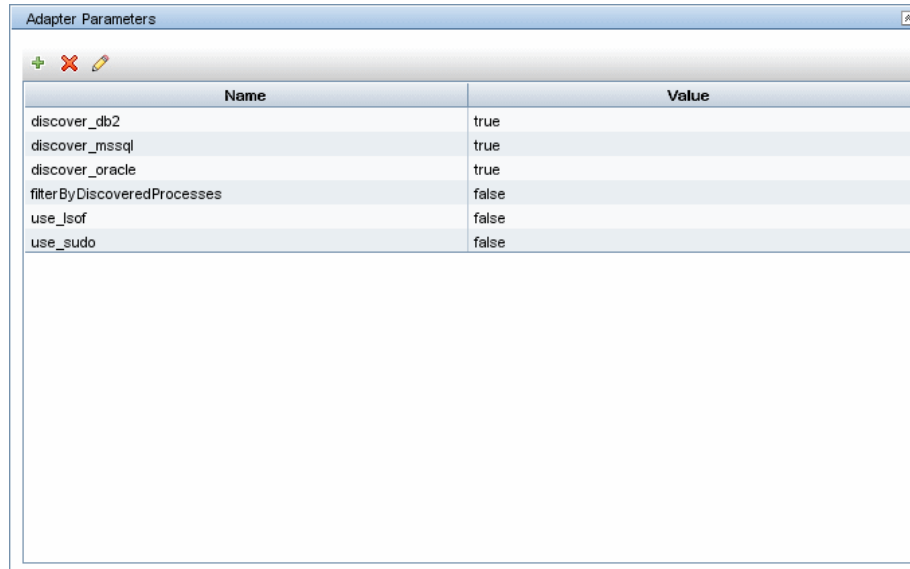
6 Trigger Query for the DB Connection by WMI Job



7 Input Query for the DB Connection by WMI Job

There is no Input Query.

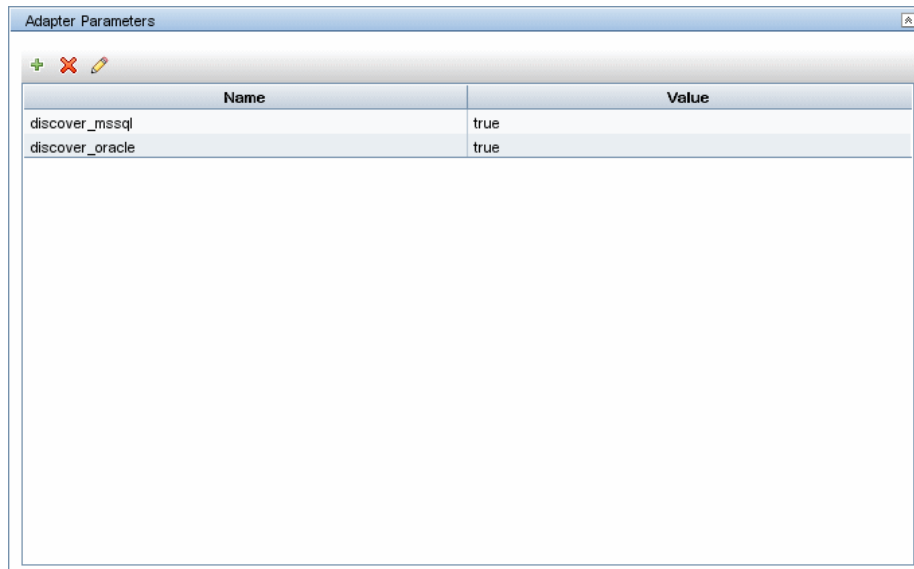
8 Adapter Parameters for the DB Connections by Shell job



Name	Value
discover_db2	true
discover_mssql	true
discover_oracle	true
filterByDiscoveredProcesses	false
use_lsof	false
use_sudo	false

- ▶ **discover_db2. true:** DFM discovers IBM DB2 database servers.
- ▶ **discover_mssql. true:** DFM discovers Microsoft SQL database servers.
- ▶ **discover_oracle. true:** DFM discovers Oracle database servers.
- ▶ **filterByDiscoveredProcesses.** This parameter should always be set to **false** because this script uses out-of-the-box process discovery on some platforms, and database processes are not included in the filters. However, since this job does not create Process CIs, setting this parameter to **false** has no adverse effects.
- ▶ **use_lsof.** Since process to port mapping on Solaris and AIX platforms requires root privileges, set this flag to **true** if the LSOF program is available on these platforms. Using LSOF does not require root privileges.
- ▶ **use_sudo.** Since process to port mapping on some UNIX platforms requires elevated privileges, set this flag to **true** if **sudo** is configured for **netstat**, **ps**, **pfiles**, **kdb**, or **lsof**.

9 Adapter Parameters for the DB Connections by WMI job



The screenshot shows a window titled "Adapter Parameters" with a table containing two rows of data. The table has two columns: "Name" and "Value". The first row has "discover_mssql" in the Name column and "true" in the Value column. The second row has "discover_oracle" in the Name column and "true" in the Value column. Above the table are three icons: a green plus sign, a red X, and a yellow pencil.

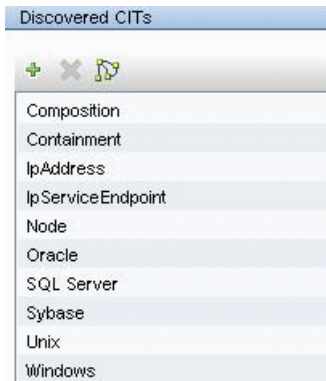
Name	Value
discover_mssql	true
discover_oracle	true

- **discover_mssql. true:** DFM discovers Microsoft SQL database servers.
- **discover_oracle. true:** DFM discovers Oracle database servers.

10 Discovered CITs

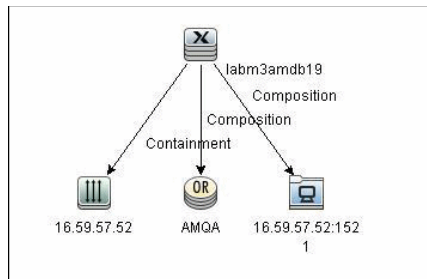
To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

► DB Connections by Shell and DB Connections by WMI

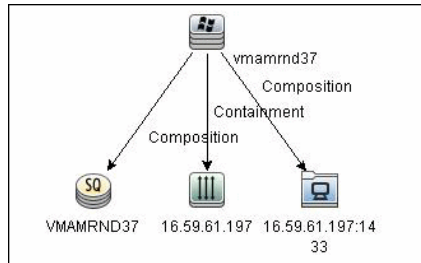


11 Sample Output

► Oracle



► Microsoft SQL



10

DB2

This chapter includes:

Tasks

- ▶ Discover IBM DB2 Databases on page 144

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 144
- "Network and Protocols" on page 144
- "Discovery Workflow" on page 144
- "Discovered CITs" on page 145
- "Topology Map" on page 145
- "Troubleshooting and Limitations" on page 146

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 **Discovery Modules > Database > DB2** module in the following order:

- DB2 Universal Database Connection by SQL
- DB2 Topology by SQL
- Databases TCP Ports

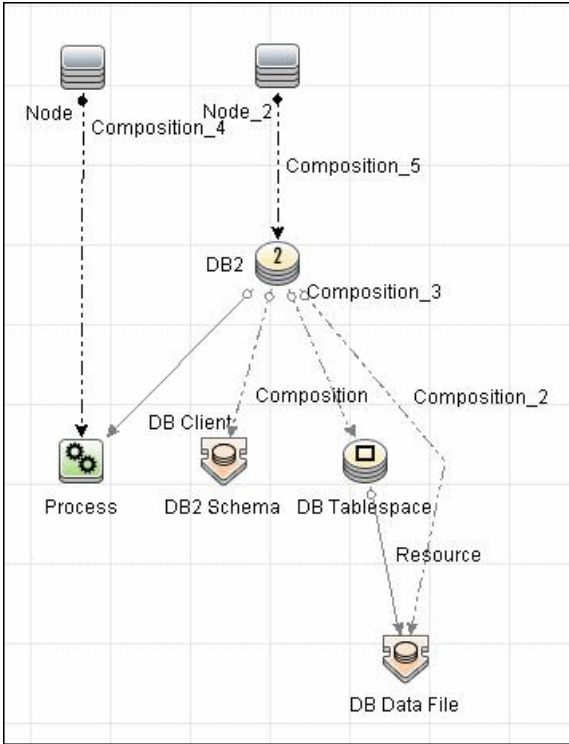
4 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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

- a** 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**
- b** Place the files in the following folder:
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\db\db2.
- c** Restart the Data Flow Probe.

11

MS-SQL

This chapter includes:

Concepts

- ▶ Discovery by OS Credentials on page 148

Tasks

- ▶ Discover Microsoft SQL Server Database Application on page 149
- ▶ Discover SQL Server by OS Credentials on page 152

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 149
- "Prerequisites" on page 149
- "Network and Protocols" on page 149
- "Adapter Parameters for the MSSQL Topology by SQL Job" on page 150
- "Discovery Workflow" on page 150
- "Discovered CITs" on page 150
- "Topology Map" on page 151
- "Troubleshooting and Limitations" on page 151

1 Supported Versions

Microsoft SQL Server 2000, 2005, 2008.

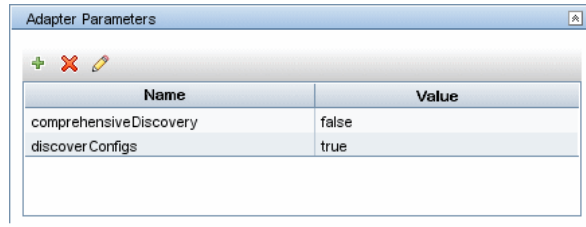
2 Prerequisites

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 Adapter Parameters for the MSSQL Topology by SQL Job



Name	Value
comprehensiveDiscovery	false
discover Configs	true

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

5 Discovery Workflow

In the Discovery Control Panel window, activate the jobs in the **Discovery Modules > Database > MS-SQL** module in the following order:

- ▶ Databases TCP Ports
- ▶ MSSQL Server Connection by SQL
- ▶ MSSQL Topology by SQL

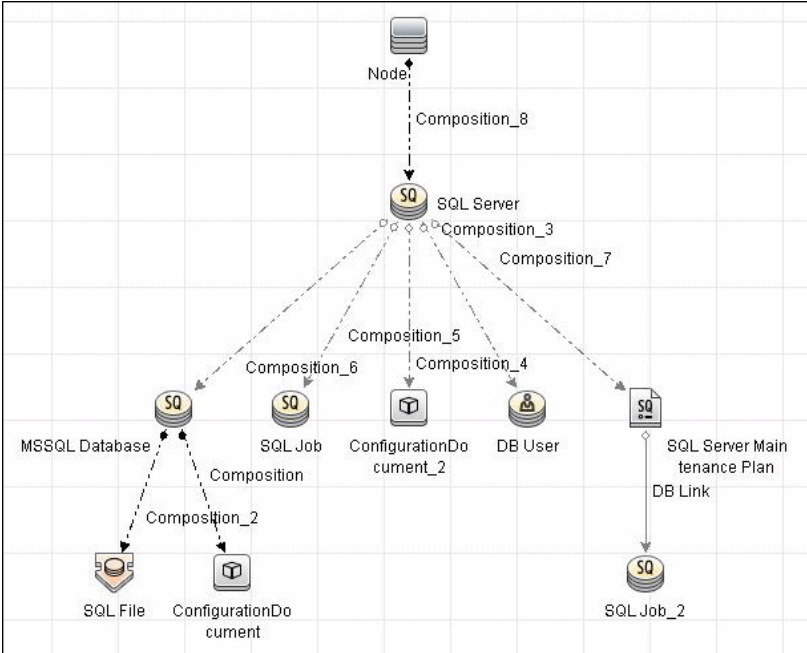
6 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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*.

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.

8 Troubleshooting and Limitations

MSSQL Server Connection by SQL. To support the connection to named instances of Microsoft SQL Server, locate and replace the following properties in the **DiscoveryProbe.properties** file.

- Replace:

```
appilog.database.sqlServer.preurl
```

with:

```
appilog.database.sqlServer.preurl=jdbc:jtds:sqlserver://%%ipaddress%%:%%pr  
otocol_port%%;instanceName=%%sqlprotocol_dbname%%;loginTimeout=%%p  
rotocol_timeout%%;logging=false;ssl=request
```

► Replace:

```
appilog.database.sqlServerNTLM.preurl
```

with:

```
appilog.database.sqlServerNTLM.preurl=jdbc:jtds:sqlserver://%%ipaddress%%:  
%%protocol_port%%;instanceName=%%sqlprotocol_dbname%%;domain=%%  
sqlprotocol_windomain%%;loginTimeout=%%protocol_timeout%%;logging=fals  
e
```

This file is located in the **C:\hp\UCMDB\DataFlowProbe\conf** directory.

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 153
- "Discovery Jobs" on page 153
- "Discovery When Host Information Is Available" on page 153
- "Discovery When Host Information Is Not Available" on page 153

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 **name** attribute was reported to UCMDB.

Data Flow Probe can report multiple SQL Server instances, each of them linked by a dependency link to its own **sqlservr.exe** process.

DFM supports SQL Server named instances.

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 When Host Information Is Available

DFM runs the following SQL command:

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

4 Discovery When Host Information Is Not Available

DFM runs the following SQL command:

```
select @@servername
```


12

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 156

Tasks

- ▶ Discover MySQL Configuration and Replication Jobs on page 157

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 157
- "Prerequisites – User Permissions" on page 158
- "Required Protocols" on page 158
- "Discovery Workflow" on page 158
- "The MySQL by Shell Job" on page 159
- "Trigger Query" on page 160
- "Configuration Item Types" on page 160
- "CIT Attributes" on page 160
- "Links" on page 161
- "Discovered CITs" on page 162
- "The MySQL Package" on page 162
- "Input Query" on page 163
- "Triggered CI Data" on page 163
- "Views – MySQL Replication Topology" on page 164
- "Limitation" on page 164

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

For credentials information, see:

- "SSH Protocol"
- "Telnet Protocol"
- "NTCMD Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

- a** Run the **Range IPs by ICMP** job to discover which of the machines in the IP range are up and running.
- b** Run the **Host Connection by Shell** job to create Shell CITs.
- c** Run any of host resources jobs to gather information about processes running on the host.
- d** 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:

```
mysql -h --verbose --help
```

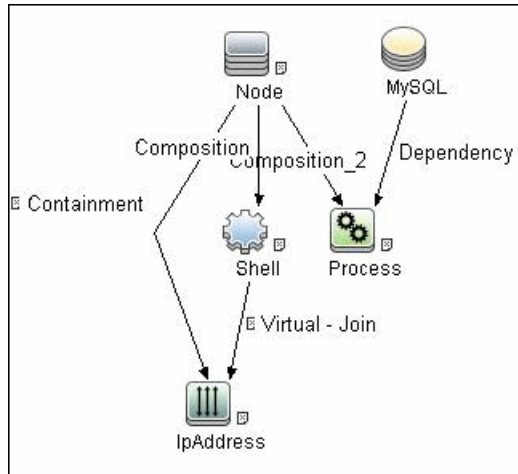
- If the job cannot find the configuration file with this command, it assumes the file is located in the default configuration file path:
 - UNIX or Linux: **/etc/my.cnf**
 - Windows: **../my.ini**
- The job tries to retrieve the attribute values from the configuration file. The job either reads the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

```
mysql -h --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 160.
- 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 **mysql -h --version** command output.
- The job creates the MySQL replication CI with appropriate attribute values and links.

6 Trigger Query



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 server ID is used in the replication job and must be unique for each server.
- **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

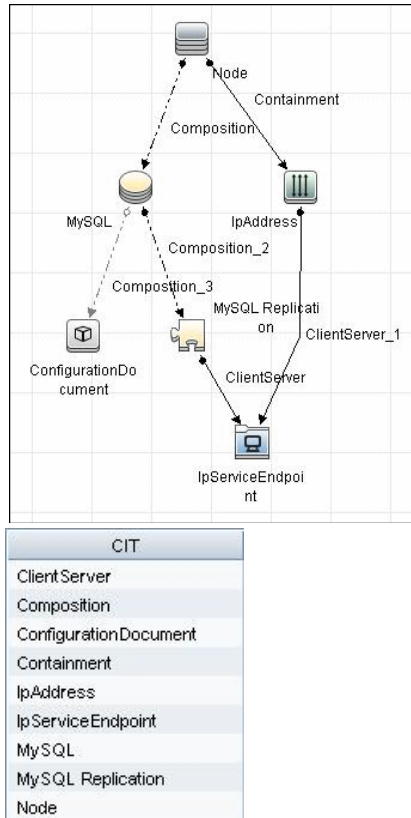
- ▶ **master_user.** A user name used when connecting to the master server.
- ▶ **master_connect_retry.** The number of seconds that the slave thread sleeps before trying to reconnect to the master, if the master goes down or the connection is lost.

9 Links

Source	Destination	Link Type	Cardinality
mysql	configfile	Composition	1..1
mysql	mysql_replication	Composition	1..1
mysql_replication	IpServiceEndpoint	ClientServer	1..1

10 Discovered CITs

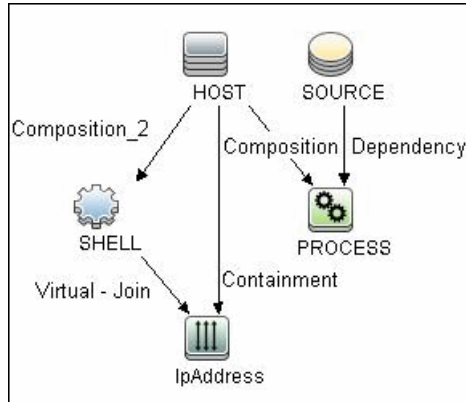
To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.



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 Universal CMDB Administration Guide*.

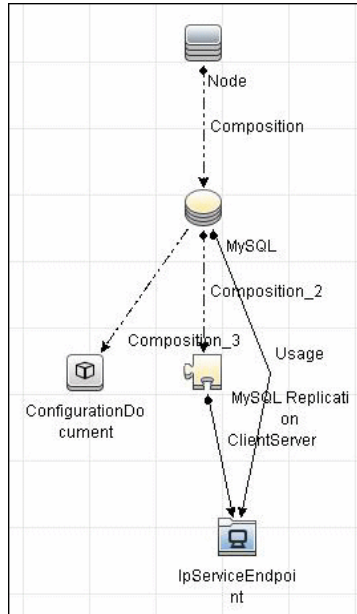
12 Input Query



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.

13

Oracle

This chapter includes:

Tasks

- ▶ Discover Oracle Databases on page 166
- ▶ Discover Oracle Real Application Cluster (RAC) on page 168

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 166
- "Prerequisites" on page 166
- "Network and Protocols" on page 166
- "Discovery Workflow" on page 167
- "Discovered CITs" on page 167
- "Topology Map" on page 168

1 Supported Versions

Oracle 8, 9, 10.

2 Prerequisites

Run **Databases TCP Ports**. Verify the user name, password, and port used by the Oracle Database Server.

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 Discovery Workflow

In the Discovery Control Panel window, activate the jobs in the **Discovery Modules > Database > Oracle** module in the following order:

- Databases TCP Ports
- Oracle Database Connection by SQL
- Oracle Topology by SQL

5 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane.

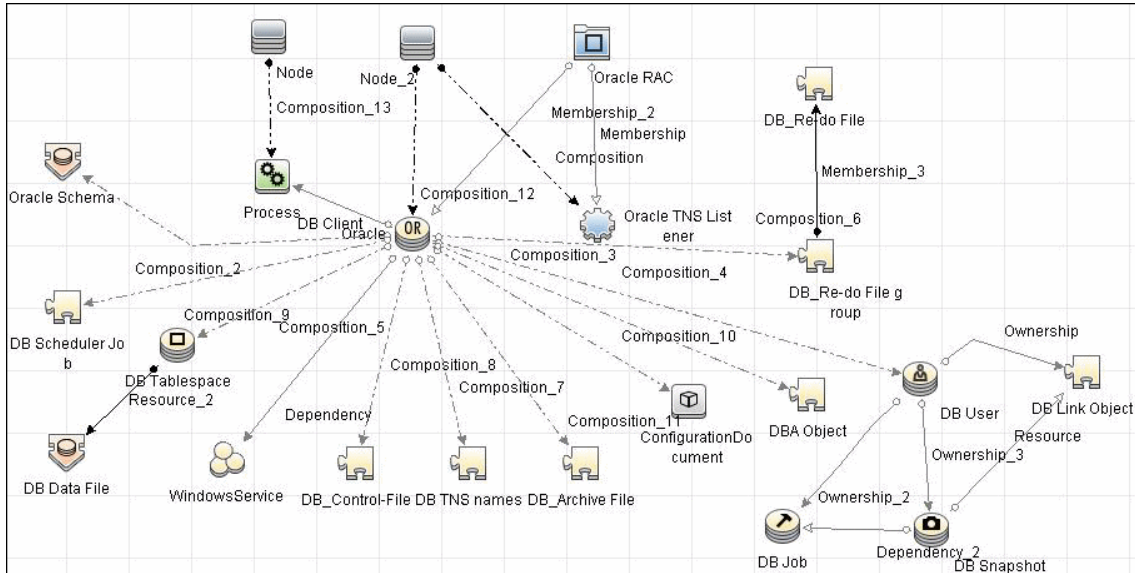
For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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

6 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 169
- ▶ "Supported Versions" on page 169
- ▶ "Prerequisites" on page 169
- ▶ "Required Protocols" on page 170
- ▶ "Discovery Workflow" on page 170

- "The Oracle Listeners by Shell Job" on page 170
- "The Oracle RAC Topology by Shell Job" on page 173
- "Topology" on page 176
- "Configuration Items" on page 176
- "The Oracle Package" on page 177
- "Oracle View" on page 178
- "Troubleshooting and Limitations" on page 178

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

For credentials information, see:

- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet Protocol"

in *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. For example, host resources and applications by Shell.

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 170.
- 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 173. For details on undiscovered elements, see "Troubleshooting and Limitations" on page 178.

6 The Oracle Listeners by Shell Job

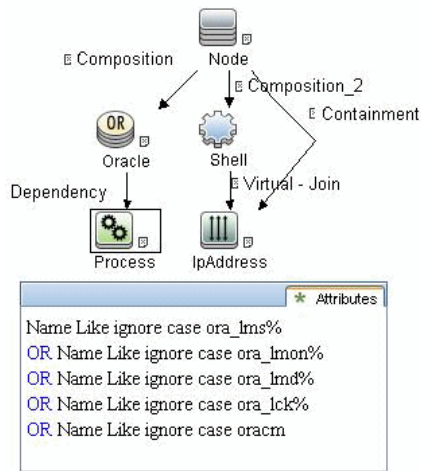
This job triggers on Oracle databases that have RAC related processes. The job:

- 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).

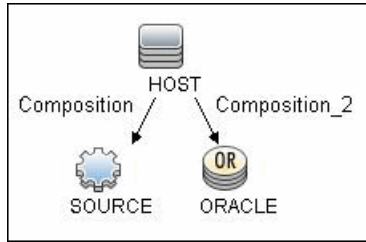
- ▶ Reads the **Oracle TNS listener** configuration file, stored in **\$ORACLE_HOME/network/admin/listener.ora**, and performs further parsing.
- ▶ Retrieves a full list of IP addresses to which this particular listener is listening.
- ▶ Checks for listener status using the **\$ORACLE_HOME/bin/lsnrctl** status.
- ▶ Retrieves known services and listener status from the output.

Trigger Query



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 Query



- **Used Scripts.** oracle_listeners_by_shell.py

Triggered CI Data

Triggered CI data	
Name	
Protocol	\${SOURCE.root_class}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

Discovered CITs

CIT
Composition
Containment
IpAddress
Node
Oracle TNS Listener
Unix

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

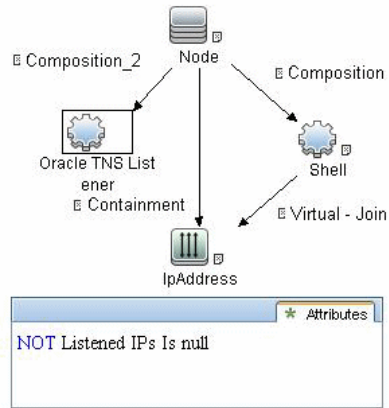
This job:

- 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.

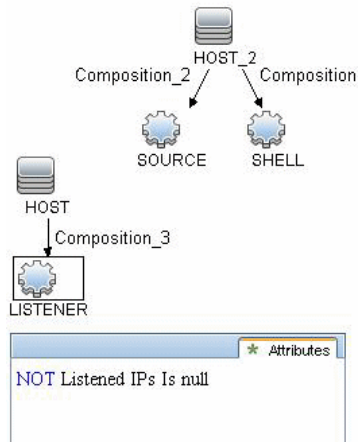
Note: Nodes are cited in the **tnsnames.ora** file by their internal IP or by their internal domain name. If the domain name appears, DFM resolves it.

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

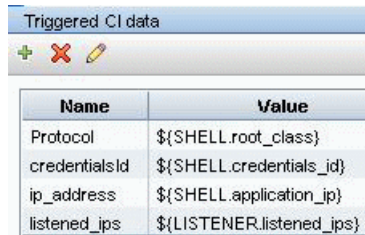
Trigger Query



Input Query



Triggered CI Data

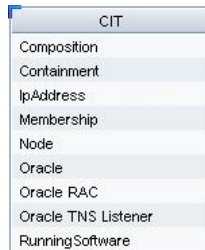


Name	Value
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
ip_address	\${SHELL.application_ip}
listened_ips	\${LISTENER.listened_ips}

Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

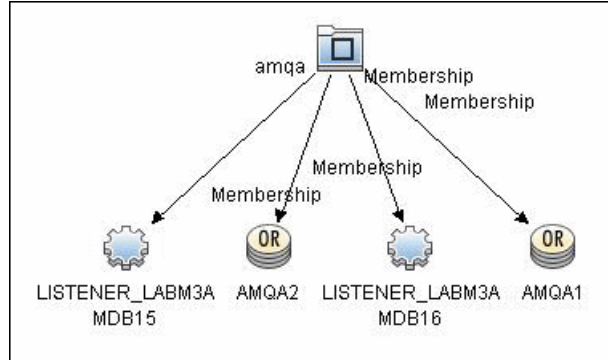


CIT
Composition
Containment
IpAddress
Membership
Node
Oracle
Oracle RAC
Oracle TNS Listener
RunningSoftware

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:**
 - **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

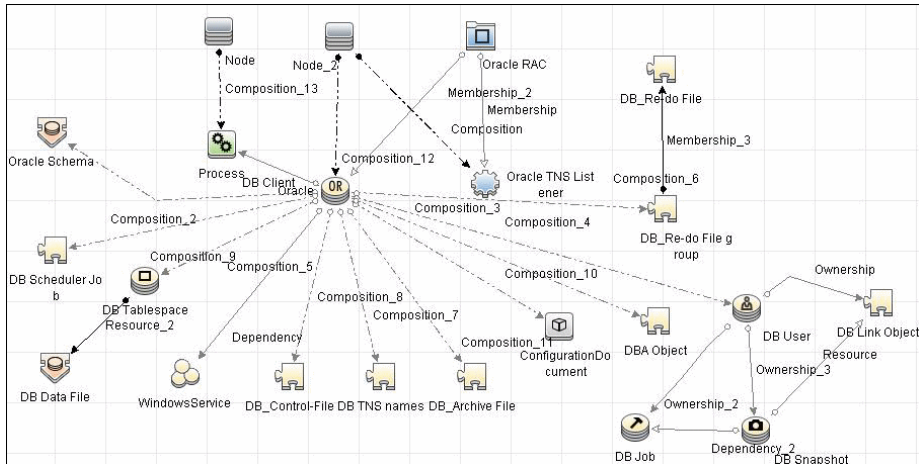
CIT	Link Type	Cardinality
Node	Composition	1.*
RAC	Membership	1.*
Process	Dependency	1.*

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.

14

Active Directory

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

This chapter includes:

Concepts

- ▶ Overview on page 182

Tasks

- ▶ Discover Active Directory Domain Controllers and Topology on page 183

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 183
- "Prerequisites" on page 183
- "Network and Protocols" on page 184
- "Discover AD Domain Controllers" on page 185
- "Discover AD Topology" on page 186
- "Active Directory Topology" on page 189

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

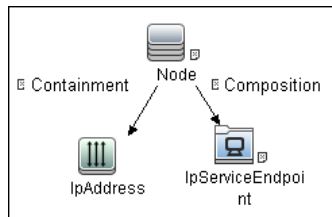
Note: User Name: if a domain is present, use **username@domain**.

4 Discover AD Domain Controllers

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

Trigger CI: IpAddress

Trigger query:



CI Attributes:

CI	Attribute Value
Source	NOT IP Probe Name Is null
IpServiceEndpoint	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 IpServiceEndpoint.
port_number	\${Service_Address.ippport_number}	The LDAP port number, retrieved from the IpServiceEndpoint.

Discovered CITs:

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

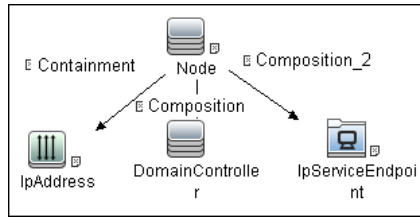
- Containment
- Composition
- DomainController
- Node
- IPAddress

5 Discover AD Topology

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

Trigger CI: DomainController

Trigger Query:



CI Attributes:

CI	Attribute Value
IpAddress	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
IpServiceEndpoint	Name Equal ignore case "ldap"

Triggered CI Data:

Name	Value	Description
application_port	\${SOURCE.application_port:NA}	The port retrieved from the IpServiceEndpoint.
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.

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**.

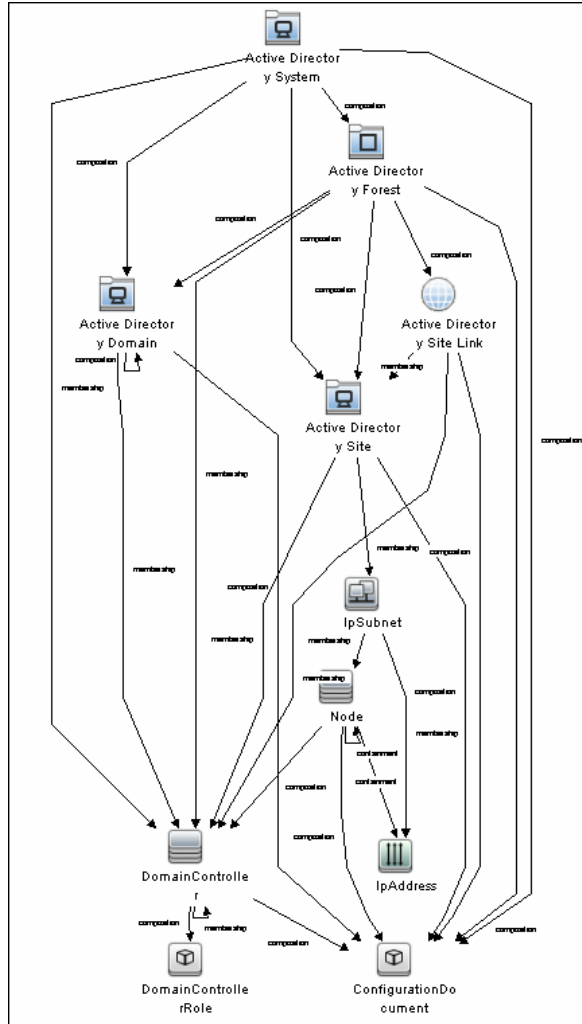
Discovered CITs

- ▶ **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.**
- ▶ **Composition.**
- ▶ **Document.**
- ▶ **DomainController**
- ▶ **DomainControllerRole**
- ▶ **Node.**
- ▶ **Membership.** Relationships between sites and subnets.
- ▶ **Network.** Available subnet objects.

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

6 Active Directory Topology



15

Microsoft Exchange

This chapter includes:

Concepts

- ▶ Overview on page 192

Tasks

- ▶ Discover Microsoft Exchange Server 2003 on page 193
- ▶ Discover Microsoft Exchange Server 2007 on page 199
- ▶ Discover Microsoft Exchange Server Topology with Active Directory on page 202

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 193
- "Prerequisites" on page 193
- "Network and Protocols" on page 194
- "Discovery Workflow" on page 194
- "Configuration Item Types" on page 195
- "Discovered CITs" on page 196
- "The Microsoft Exchange Server Package" on page 196
- "Queries" on page 197
- "Views" on page 197
- "Topology Map" on page 197
- "Troubleshooting and Limitations" on page 198

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 198.

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 198.)

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.

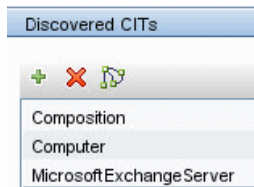
c Microsoft Exchange Resource

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

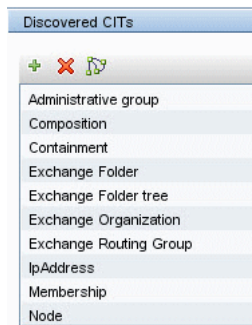
- ▶ **Exchange folder.** This CIT represents the public folders available in the Exchange organization. A public folder may be organized in a hierarchical structure, that is, one public folder may contain another public folder.
- ▶ **Exchange folder tree.** This CIT provides information about public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

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

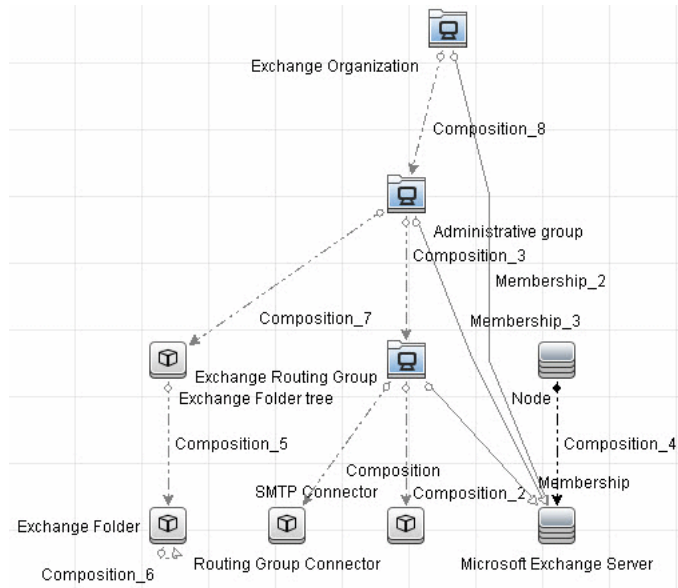
9 Views

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

10 Topology Map

MS Exchange 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 query. 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 193.

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 200
- "Supported Versions" on page 200
- "Network and Protocols" on page 200
- "Discovery Workflow" on page 200

- "The Microsoft Exchange Server Package" on page 201
- "Discovered CITs" on page 201
- "Topology Maps" on page 202

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

For credentials information, see:

- "NTCMD Protocol"
- "LDAP Protocol"

in *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:

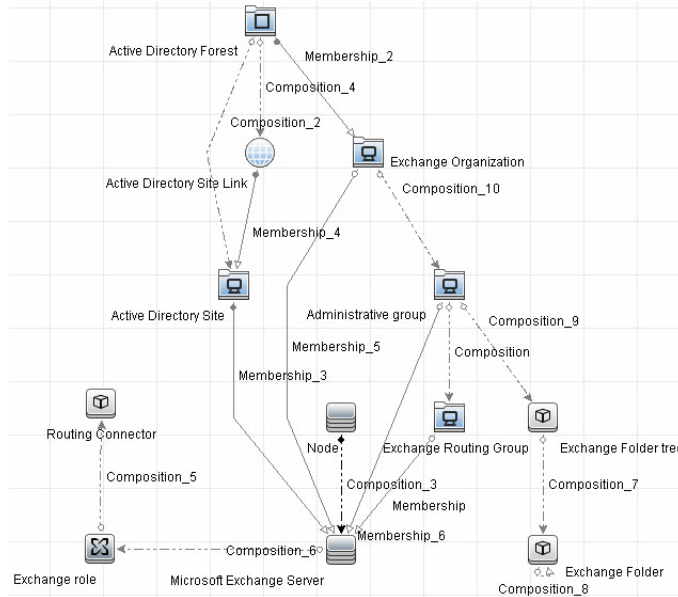
- ▶ **Exchange Organization.** This CIT represents the top-level Exchange system. For example, if an organization uses the Exchange solution, all the Exchange components are linked to a single Exchange Organization CI.
- ▶ **Microsoft Exchange Server.** This CIT is inherited from the RunningSoftware 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.

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

7 Topology Maps

The following maps illustrate Microsoft Exchange Server 2007 topology.

MS Exchange Connection by NTCMD:



MS Exchange 2007 Topology:

Discover Microsoft Exchange Server Topology with Active Directory

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

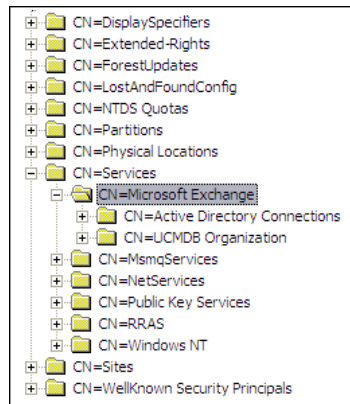
This section explains how DFM discovers Exchange by utilizing the tight integration between Exchange and AD. DFM runs jobs to discover Exchange elements in the topology that are available only through AD.

This task includes the following steps:

- "Overview" on page 204
- "Prerequisites – Set Permissions" on page 206
- "Prerequisites – Discover a Domain Controller" on page 206
- "Supported Versions" on page 207
- "Network and Protocols" on page 207
- "Discovery Workflow" on page 207
- "The Microsoft Exchange Server Package" on page 207
- "Additional CITs" on page 207
- "Deprecated CITs" on page 208
- "Modified CITs" on page 208
- "Discovered CITs" on page 208
- "Trigger Query" on page 209
- "Trigger CI" on page 209
- "CI Attributes" on page 209
- "Adapters" on page 210
- "Topology Maps" on page 210
- "Troubleshooting and Limitations" on page 210

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 206.

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**.

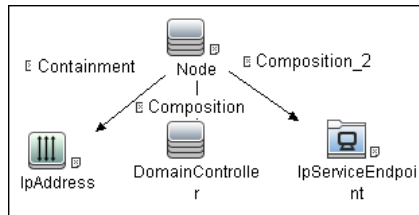
11 **Discovered CITs**

- ▶ Active Directory Forest
- ▶ Active Directory Site
- ▶ Active Directory System
- ▶ Administrative Group
- ▶ Containment
- ▶ Composition
- ▶ Exchange Folder
- ▶ Exchange Folder Tree
- ▶ Exchange Organization

- Exchange Routing Connector
- Exchange role
- Host
- IPAddress
- Membership
- Microsoft Exchange Server
- Routing Group Connector
- Routing group
- SMTP Connector

12 Trigger Query

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



13 Trigger CI

DomainController

14 CI Attributes

CI	Attribute Value
IpAddress	NOT IP Probe Name Is null
DomainController	NOT Reference to the credentials entry dictionary Is null AND NOT Application IP Is null
IpServiceEndpoint	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 202.
- ▶ For the Microsoft Exchange Server 2003 topology view, see "Topology Map" on page 197.

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.

16

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 212

Reference

- ▶ Topology Discovery Methodology on page 216
- ▶ Added Entities on page 226
- ▶ Removed Entities on page 227

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 212
- "Discovery Workflow" on page 212
- "Scripts" on page 213
- "Trigger Queries" on page 214
- "Input Queries" on page 214
- "Performance" on page 215

1 Supported Versions

MS MQ version 3.0 or later

2 Discovery Workflow

Activate the jobs in the following order:

- a Host Connection by Shell**
- b 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 (the physical machine on which the MS MQ is installed) is a member of the domain.

c Active Directory Connection by LDAP

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

d Microsoft Message Queue Topology by NTCMD

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

e 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 216.

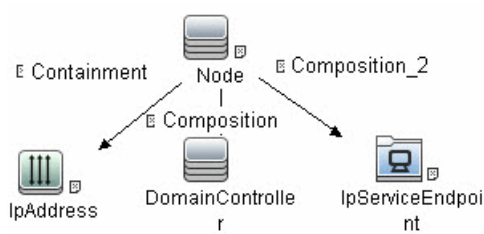
3 Scripts

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

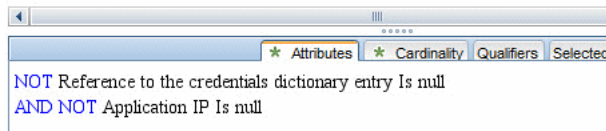
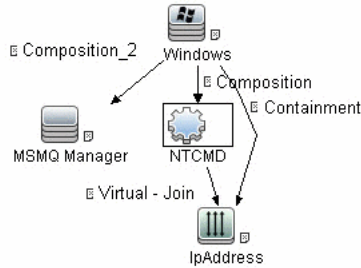
Script	Description
ntcmd_msmq.py	Main script for the Microsoft Message Queue Topology by NTCMD job
ldap_msmq.py	Main script for the Microsoft Message Queue Topology by LDAP job
plugin_microsoft_mq.py	Shallow plug-in for MS MQ Manager discovery (Adapter Management > Discovery Packages > Host_Resources_Basic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

4 Trigger Queries

Microsoft Message Queue Topology by LDAP:

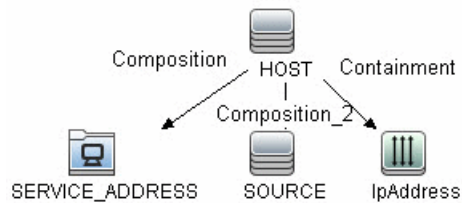


Microsoft Message Queue Topology by NTCMD:

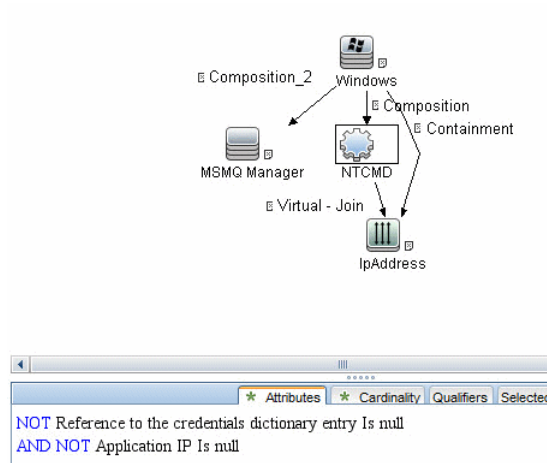


5 Input Queries

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 216
- "Microsoft Message Queue Topology by NTCMD Job" on page 218
- "Microsoft Message Queue Topology by LDAP Job" on page 225

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
Enterpriseld 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\qs
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
00051200000001010000000000005120000000
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):

```
"^[\\]*\s*(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]{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+(0x\d+).**"
```

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	Msmqqueue
CI Type	Msmqroutinglink
CI Type	Msmqrule
CI Type	Msmqtrigger
Attribute type definition	MessageProcessingTypeEnum
Type definition	MsMqManagerInstallationType
Type definition	MsMqQueueTypeEnum
Link	clientserver.msmqmanager.msmqmanager

Entity Type	Changed Entity
Link	containment.msmqroutinglink.mqqueuemanager
Link	containment.msmqroutinglink.msmqmanager
Link	composition.activedirectoryforest.msmqroutinglink
Link	composition.msmqqueue.msmqtrigger
Link	membership.msmqroutinglink.activedirectorysite
Link	usage.msmqtrigger.msmqrule
Job	Microsoft Message Queue Topology by LDAP
Job	Microsoft Message Queue Topology by NTCMD

Removed Entities

In version 9.01, the MQ (Microsoft Message Queue) model has been changed and the following resources are no longer available.

The following CITs are deprecated:

- mqaliasq, display name: IBM MQ Queue Alias
- mqalias, display name: IBM MQ Alias
- mqchannelof, display name: IBM MQ Channel Of
- mqchannel, display name: IBM MQ Channel
- mqchclntconn, display name: IBM MQ Client Connection Channel
- mqchclusrcvr, display name: IBM MQ Cluster Receiver Channel
- mqchclusdr, display name: IBM MQ Cluster Sender Channel
- mqchrcvr, display name: IBM MQ Receiver Channel
- mqchrqstr, display name: IBM MQ Requester Channel
- mqchsdr, display name: IBM MQ Sender Channel
- mqchsvrconn, display name: IBM MQ Server Connection Channel
- mqchsvr, display name: IBM MQ Sender Channel

- mqcluster, display name: IBM MQ Cluster
- mqmqichannel, display name: IBM MQ MQI Channel
- mqmqilink, display name: IBM MQ
- mqmsgchannel, display name: IBM MQ Message Channel
- mqmsglink, display name: IBM MQ Message
- mqmsgreceiverchannel, display name: IBM MQ Message Receiver Channel
- mqmsgsenderchannel, display name: IBM MQ Messenger Sender Channel
- mqqueuelocal, display name: IBM MQ Local Queue
- mqqueuemanager, display name: IBM MQ Queue Manager
- mqqueueremote, display name: IBM MQ Remote Queue
- mqqueue, display name: IBM MQ Queue
- mqrepository, display name: IBM MQ Repository
- mqresolve, display name: IBM MQ Resolve
- mqxmitq, display name: IBM MQ Transmission Queue
- webspheremq, display name: IBM WebSphere MQ

The following resources have been removed:

- Enrichment rule: Create_Msg_Channel_Link_Host
- Enrichment rule: Create_Msg_Channel_Link_IP
- Enrichment rule: Create_RemoteQueue_Link
- Enrichment rule: Host_Depend_By_MQ
- View: MQ_All_Objects
- View: MQ_Channels
- View: MQ_Clusters
- View: MQ_Network_Objects
- View: MQ Queue Map
- TQLs: All TQLs corresponding to the above Enrichment rules and Views

17

SAP

This chapter includes:

Concepts

- ▶ SAP Discovery Overview on page 230

Tasks

- ▶ Discover SAP ABAP on page 233
- ▶ Discover SAP Solution Manager on page 238
- ▶ Discover SAP Java on page 241
- ▶ Topology Map on page 244

Concepts

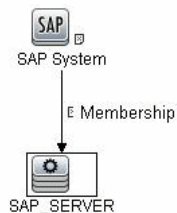
SAP Discovery Overview

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*.

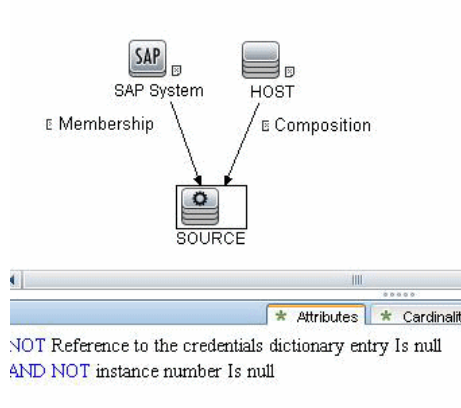
The Trigger CI for the following jobs is **SAP ABAP Application Server**:

- ▶ SAP Solution Manager Topology by SAP JCO
- ▶ SAP Solution Manager by SAP JCO

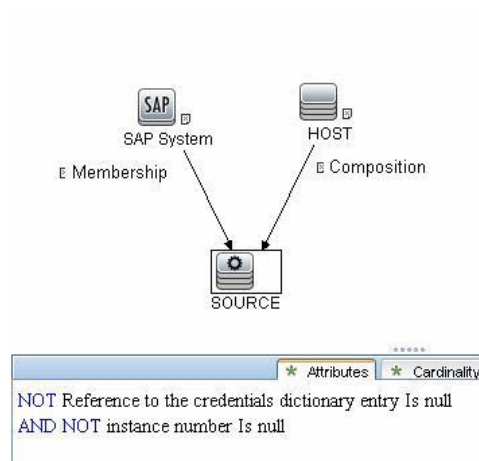


Attributes	Cardinality	Qu
NOT Reference to the credentials dictionary entry Is null		
AND NOT instance number Is null		

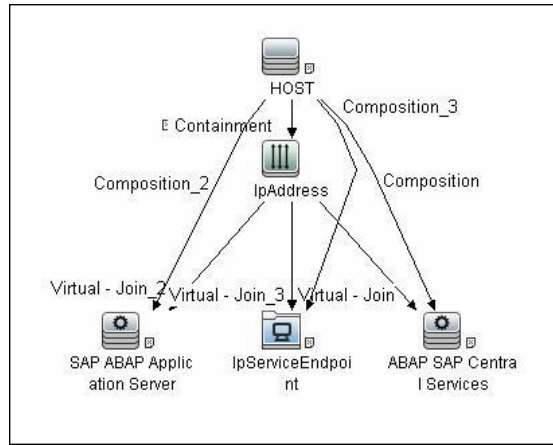
► SAP Applications by SAP JCO



► SAP ABAP Topology by SAP JCO

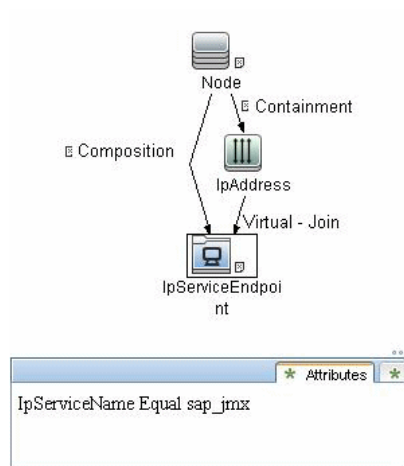


► SAP ABAP Connection by SAP JCO



The Trigger CI for the following job is **IpAddress**:

► SAP Java Topology by SAP JMX



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 233
- "Prerequisites – Install Java Connectors" on page 234
- "Network and Protocols" on page 235
- "Discovery Workflow" on page 235
- "Configure Adapter Parameters" on page 237

1 Supported Versions

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

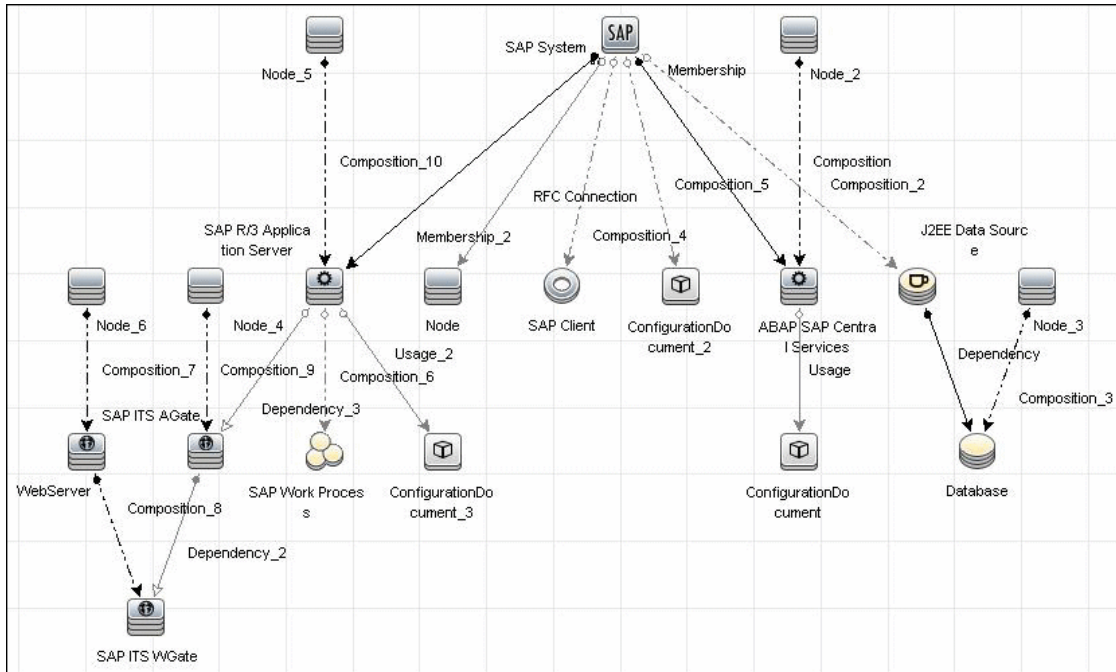
SAP JCo. Version 2.x (recommended). Note that DFM can discover SAP as long as the default SAP JCo provided with DFM is the correct version. If you are running an older version of SAP JCo, DFM may not be able to connect to SAP version 6.x.

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\dll 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\dll** 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\SysWOW64**).

Place the **msvcp71.dll** and **msvcr71.dll** drivers under the Windows installation folder (for example, **C:\windows\SysWOW64**).

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. For credentials information, see:

- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet Protocol"
- "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 or Range IP by nmap, Host Connection By Shell).
 - **Host Resources and Applications** (Host Resources and Applications by Shell). This job discovers SAP running software and processes.

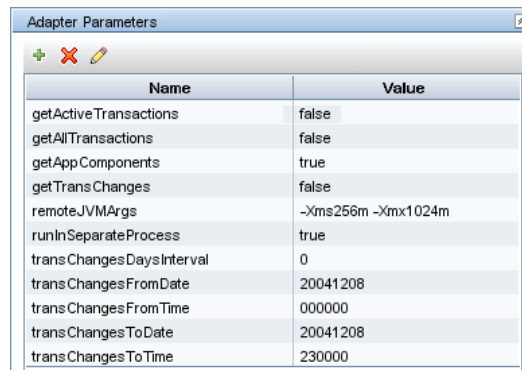
- ▶ **Network – Advanced** (TCP Ports). Also, activate the SAP System by Shell job. This job discovers SAP J2EE Central Services and a SAP system without SAP J2EE credentials.
- ▶ **Web Servers – Basic** (WebServer Detection using TCP Ports). If the SAP system has an ITS configuration, to discover the 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 237.
 - ▶ **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 238.

- 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 Window" 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.

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 239
- ▶ "Supported Versions" on page 239
- ▶ "Network and Protocols" on page 239
- ▶ "Discovery Workflow" on page 239

- "Discovered CITs" on page 240
- "Topology Map" on page 241

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.

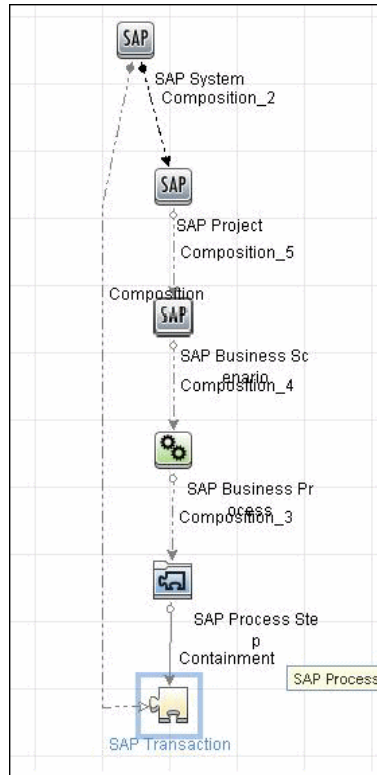
5 Discovered CITs

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

Discovered CITs
  
ABAP SAP Central Services
Composition
ConfigurationDocument
Containment
Database
Dependency
IpAddress
J2EE SAP Central Services
JDBC Data Source
Membership
Node
SAP ABAP Application Server
SAP Client
SAP J2EE Application Server
SAP System
Usage

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 242

- "Network and Protocols" on page 243
- "Discovery Workflow" on page 244
- "Topology Map" on page 244

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.

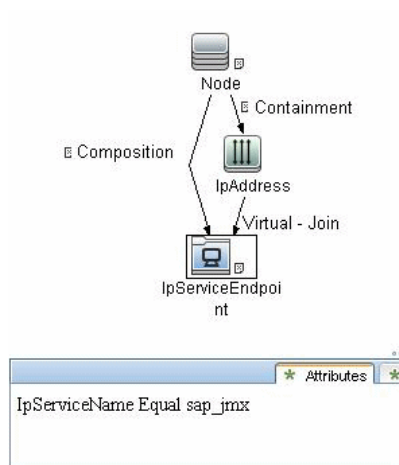
- b** Add the **com_sap_pj_jmx.jar** file to the **C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\j2ee\sap** directory on the Data Flow Probe machine:

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:

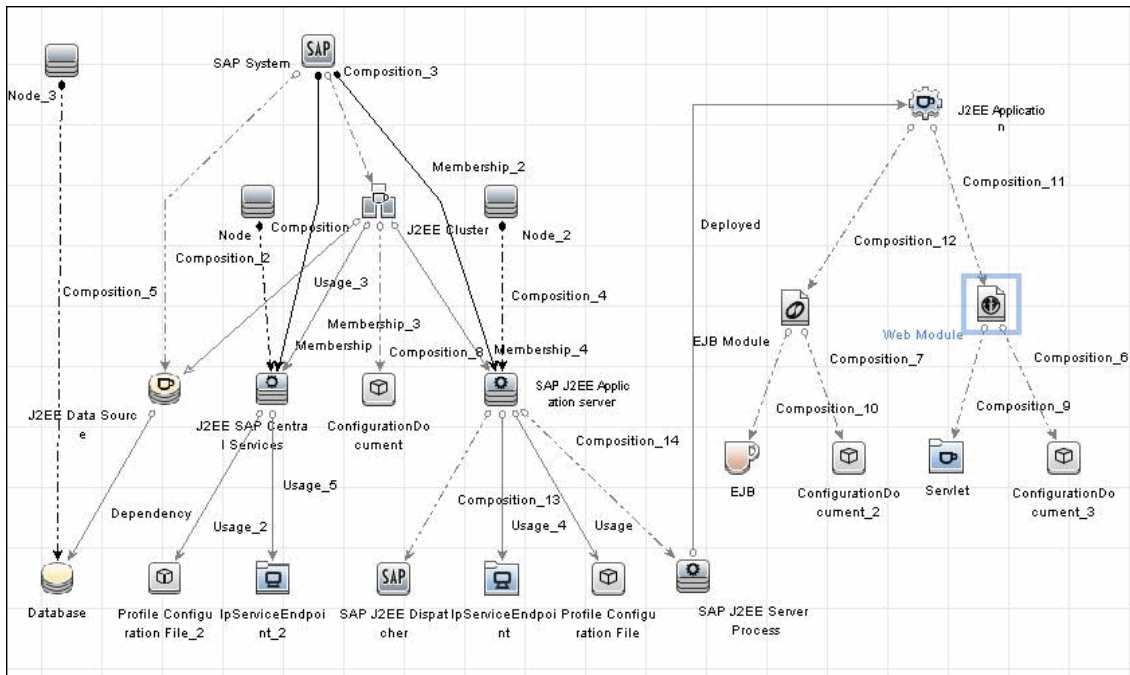
- **SAP JMX.** For credentials information, see "SAP JMX Protocol" in *HP Universal CMDB Data Flow Management Guide*.

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 Topology Map



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.

18

Siebel

This chapter includes:

Concepts

- ▶ Overview on page 248

Tasks

- ▶ Discover Siebel Topology on page 249

Troubleshooting and Limitations on page 260

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 249
- "Network and Protocols" on page 250
- "Trigger Queries" on page 251
- "Discovery Workflow" on page 254
- "Discovered CITs" on page 255
- "Topology Map – Siebel Topology View" on page 258
- "Topology Map – Siebel Web Topology View" on page 259

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 Protocol"
- "NTCMD Protocol"
- "Siebel Gateway Protocol"
- "SAP Protocol"

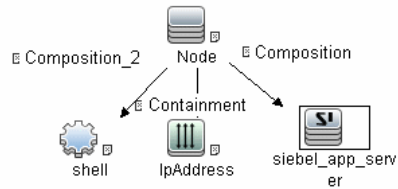
in *HP Universal CMDB Data Flow Management Guide*.

Set up the following protocols for the UNIX platform:

- "SSH Protocol"
- "Telnet Protocol"
- "Siebel Gateway Protocol"

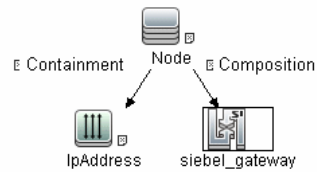
3 Trigger Queries

- The Siebel Application Server Configuration job:



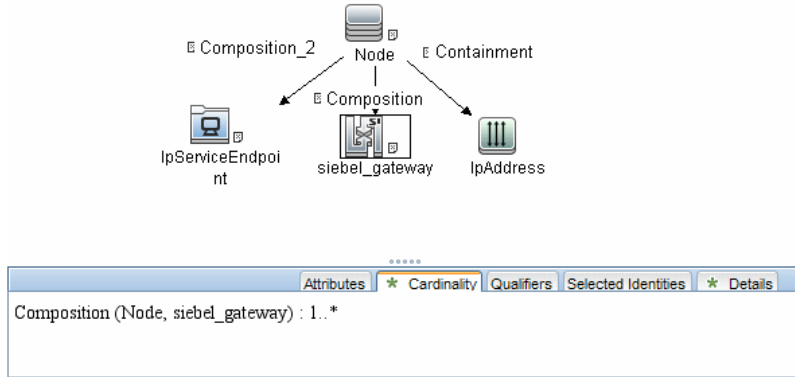
Attributes	* Cardinality	Qualifiers	Selected Identities	* Details
Composition (Node, siebel_app_server) : 1..*				

- The Siebel Application Servers job:

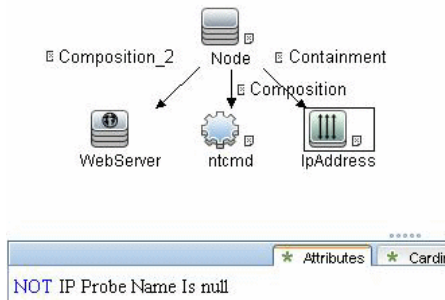


* Attributes	* Cardinality	Qualifiers	Selected Identities	* Details
NOT Application Username Is null				

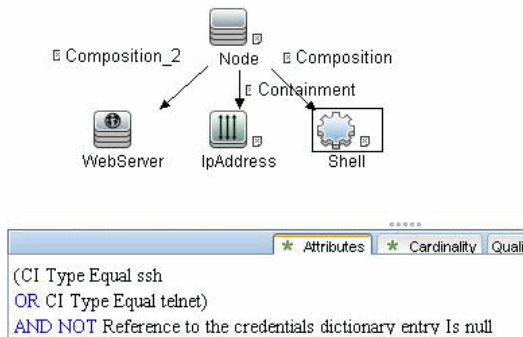
► The Siebel Gateway Connection job:



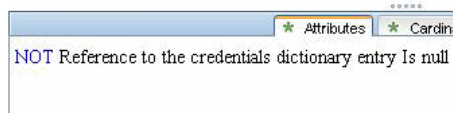
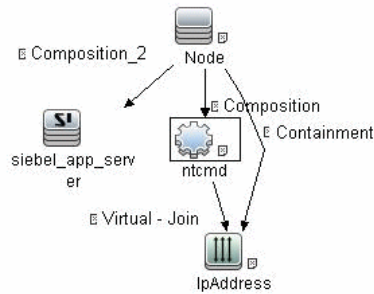
► The Siebel Web Applications by NTCMD job:



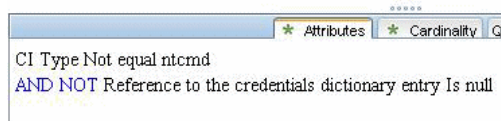
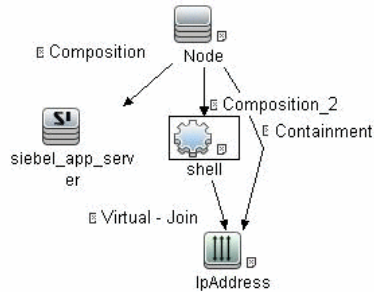
► The Siebel Web Applications by TTY job:



- The Siebel DB by NTCMD job:



- The Siebel DB by TTY job:



4 Discovery Workflow

- a** 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 the *HP Universal CMDB Modeling Guide*.
- b** 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)
- c** 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)
- d** To discover Siebel, activate all the jobs in the **Application – Siebel** module.

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

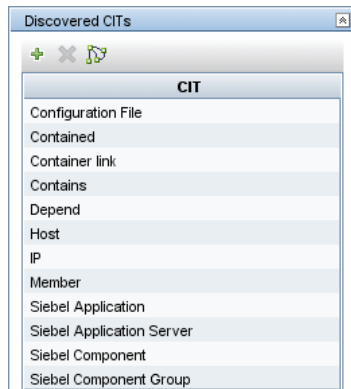
Siebel_Route_WebApp_To_Component. Builds the route between Siebel Web Application CIs and Siebel Component CIs.

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.

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

5 Discovered CITs

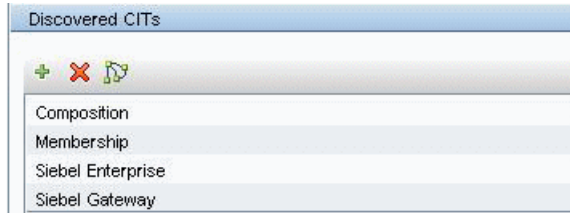
SIEBEL_DIS_APP_SERVERS:



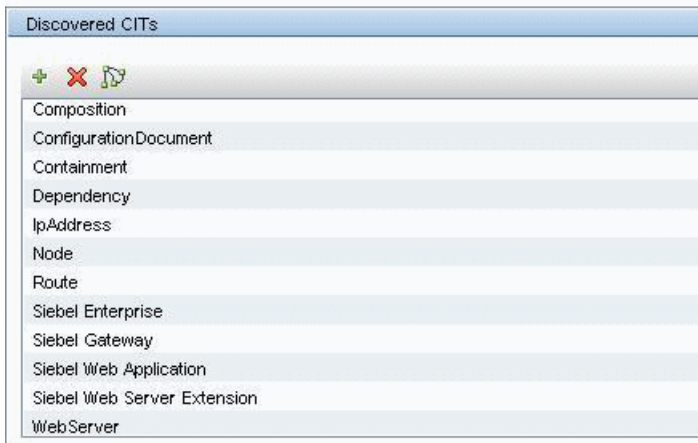
SIEBEL_DIS_APP_SERVER_CONFIG:



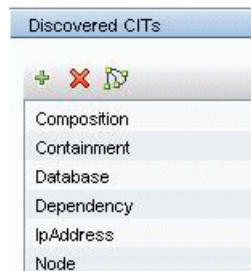
SIEBEL_DIS_GATEWAY_CONNECTION_(GTWY)



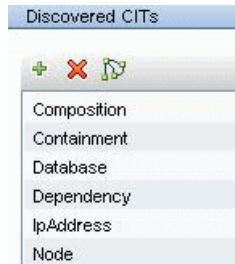
SIEBEL_DIS_WEBAPPS_UNIX:



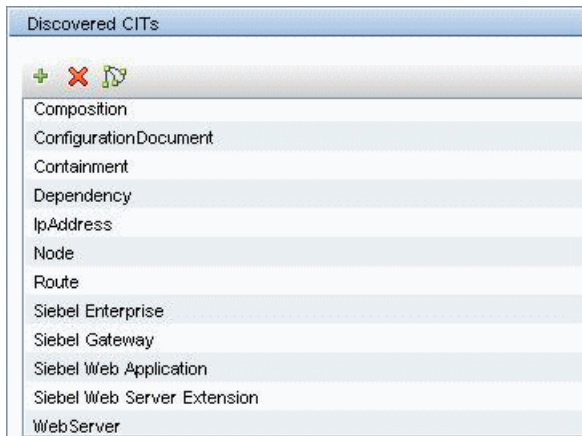
SIEBEL_DIS_DB_UNIX



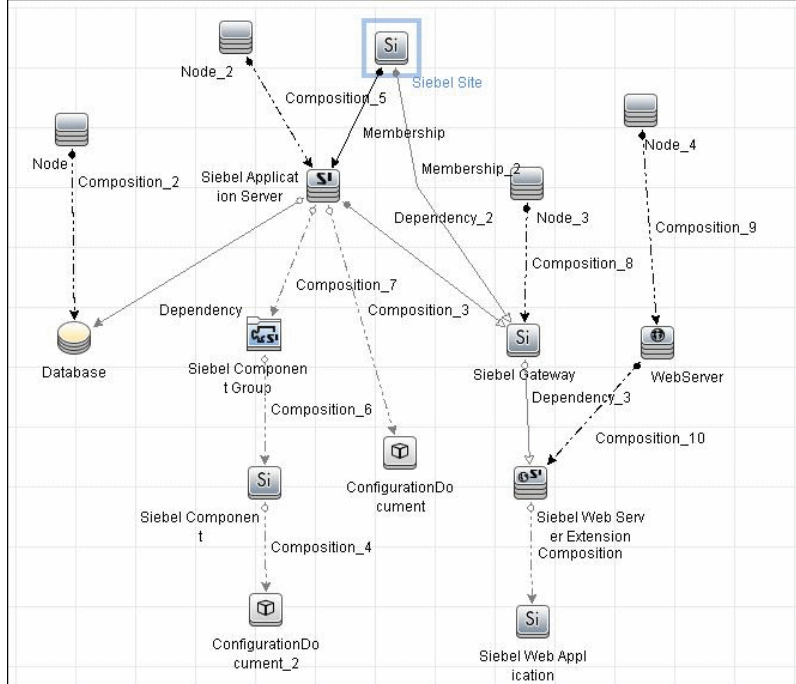
SIEBEL_DIS_DB_NT



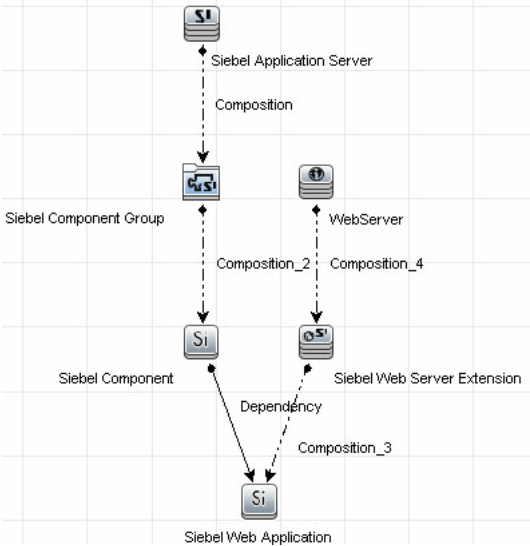
SIEBEL_DIS_WEBAPPS_NT



6 Topology Map – Siebel Topology View



7 Topology Map – Siebel Web Topology View



Reference

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.

19

UDDI Registry

This chapter includes:

Concepts

- ▶ Overview on page 262

Tasks

- ▶ Discover UDDI Processes on page 263

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 263
- "Network and Protocols" on page 263
- "Discovery Workflow" on page 263
- "Discovery Workflow – Optional" on page 264
- "Topology Map" on page 264

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.

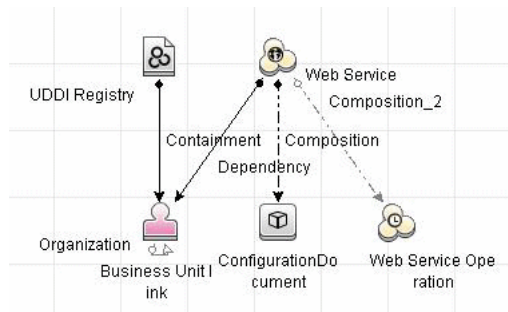
4 Discovery Workflow – Optional

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_discovery 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**:



20

WebSphere MQ

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

This chapter includes:

Concepts

- ▶ Overview on page 266

Tasks

- ▶ Discover WebSphere MQ on page 269

Reference

- ▶ Discovered CITs on page 273
- ▶ Relationships on page 276
- ▶ Enrichment Rule on page 279
- ▶ Views and Reports on page 280

Troubleshooting and Limitations on page 281

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 queries 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 269
- "Network and Protocols" on page 269
- "Package Deployment" on page 270
- "Discovery Workflow" on page 270
- "Adapter Parameters" on page 271
- "Discovered CITs" on page 272

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

For credentials information, see:

- "NTCMD Protocol"
- "SSH Protocol"
- "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

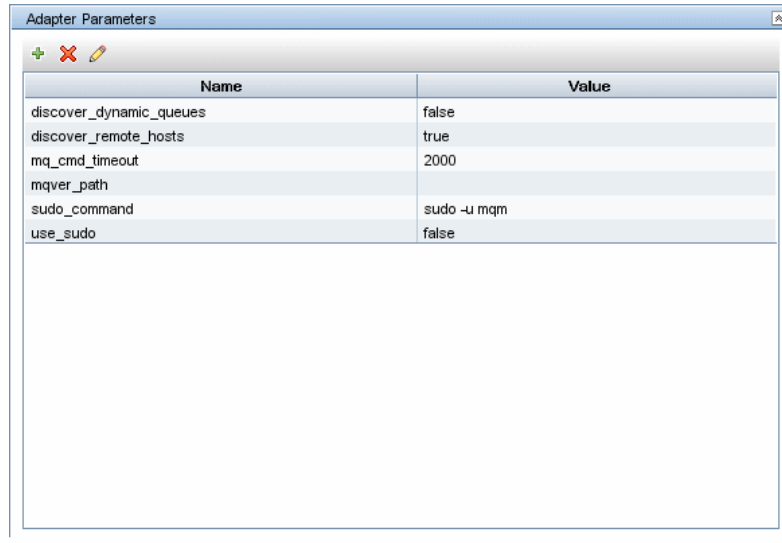
- a** Deploy the WebSphere MQ package. For details, see "Deploy a Package" in the *HP Universal CMDB Administration Guide*.
- b** Populate the appropriate SSH, Telnet, or NTCMD protocol. For details, see "Network and Protocols" on page 269.
- c** Verify that all WebSphere MQ server IP addresses are within 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 Pane" 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

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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: Node 	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.

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.



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.

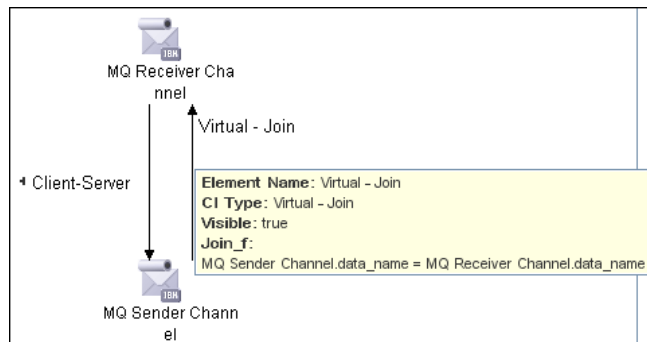
Link	End1	End2	Cardinality	Description
Realization	IBM MQ Alias Queue	IBM MQ Remote Queue	1..*	Indicates a strong dependency between an MQ Alias Queue and a Remote Queue for which it is an alias.
Realization	IBM MQ Alias Queue	IBM MQ Alias Queue	1..*	Indicates a strong dependency between an MQ Alias 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.
Membership	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
Membership	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.
Membership	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.
Membership	IBM MQ Namelist	IBM MQ Channel	1..*	Indicates that the MQ Channel contains the name of the MQ Namelist in its CLUSNL parameter.
Membership	IBM MQ Namelist	IBM MQ Queue	1..*	Indicates that the MQ Queue contains the name of the MQ Namelist in its CLUSNL parameter.

Link	End1	End2	Cardinality	Description
Usage	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.
Usage	IBM MQ Remote Queue	IBM MQ Transmit Queue	1..*	Indicates a remote queue using a transmission queue for communication.
Usage	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 query 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.

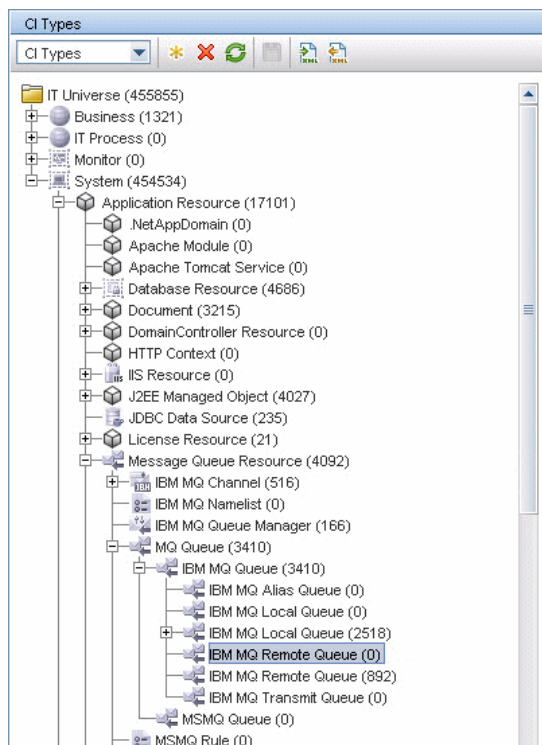
MQ Channel Communication. This view displays client-server communication between MQ Channels and queues used by the channels.

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 271.
- ▶ If the discovery results appear incomplete, try increasing the value of the **mq_cmd_timeout** parameter. For details, see "Adapter Parameters" on page 271.
- ▶ 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 **mqqueuelocal** and **mqqueueremote** 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:.

21

JBoss

This chapter includes:

Concepts

- ▶ JBoss Discovery Overview on page 284

Tasks

- ▶ Discover JBoss by JMX on page 285
- ▶ Discover JBoss by Shell on page 287

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 285
- "Supported Versions" on page 285
- "Network and Protocols" on page 285
- "Discovery Workflow" on page 286
- "Adapter Parameters for JBoss by JMX" on page 286
- "Discovered CITs" on page 286

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

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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 288
- "Supported Versions" on page 288
- "Network and Protocols" on page 288

- "Discovery Workflow" on page 288
- "Discovered CITs" on page 291

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

For credentials information, see:

- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet 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, 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:

- 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)

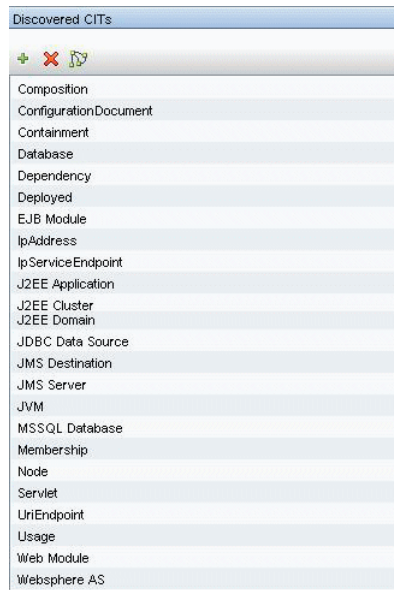
- the **xxx-ds.xml** files (for data sources)

- **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:



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.

22

WebLogic

This chapter includes:

Tasks

- ▶ Discover J2EE WebLogic by JMX on page 294
- ▶ Discover J2EE WebLogic by Shell on page 298

Troubleshooting and Limitations on page 302

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 294
- "Supported Versions" on page 294
- "Network and Protocols" on page 295
- "Discovery Workflow" on page 295
- "Adapter Parameters for J2EE Weblogic by JMX" on page 295
- "Trigger Queries" on page 296
- "Discovered CITs" on page 297

1 Prerequisites

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

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

- a** In the Discovery Control Panel window, run the **Discovery Modules > Network Discovery > Basic > Range IPs by ICMP** job.
- b** Run the **Discovery Modules > J2EE Application Servers > WebLogic > J2EE TCP Ports** job.
- c** Run the **J2EE Weblogic Connections by JMX** job.

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

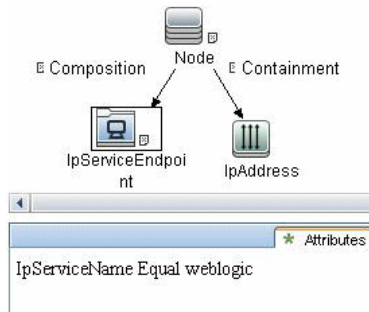
5 Adapter Parameters for J2EE Weblogic by JMX

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

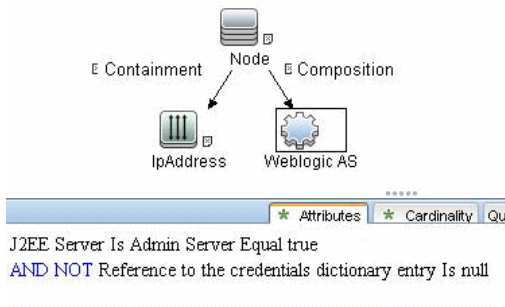
- **deploymentDescriptors.** **True:** retrieves deployment descriptors of J2EE applications, EJB modules, and Web modules.
- **discoverAppResources.** **True** or missing (as previously): a full application deployment discovery is performed. **False:** only the application is discovered without all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

6 Trigger Queries

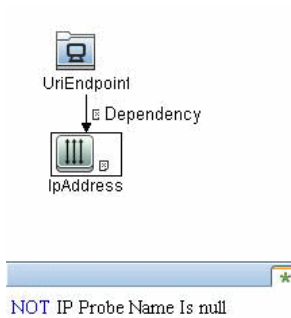
- ▶ The J2EE Weblogic Connections by JMX job:



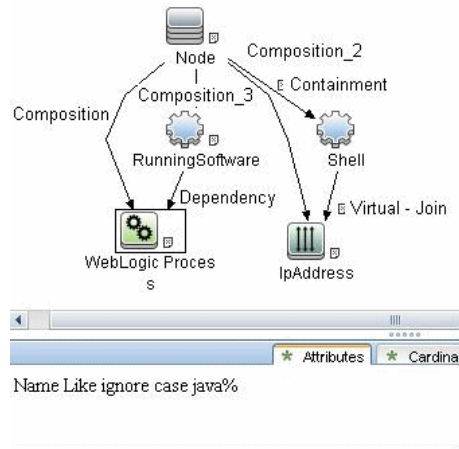
- ▶ The J2EE Weblogic by JMX job:



- ▶ The WebServices by URL job:



► The J2EE Weblogic by Shell job:

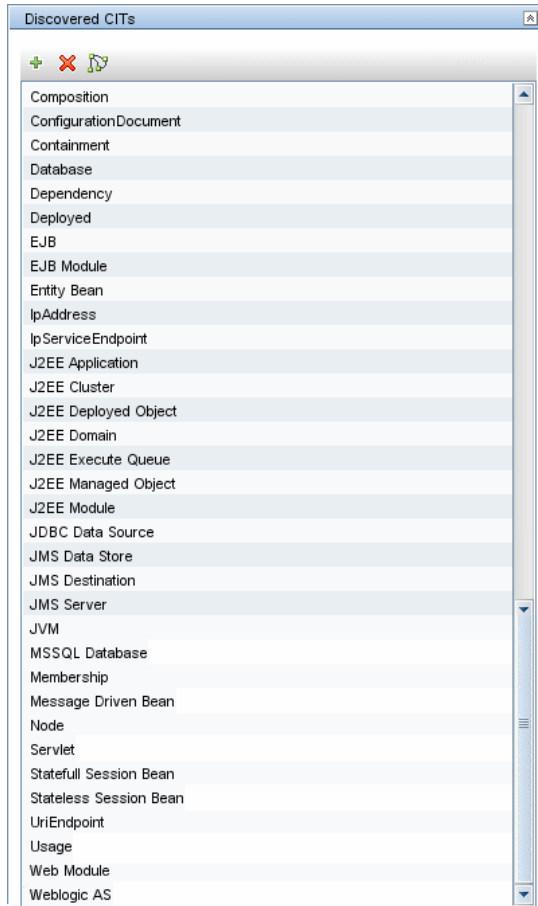


7 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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 299
- "Supported Versions" on page 299
- "Network and Protocols" on page 299
- "Discovery Workflow" on page 300
- "Discovered Elements" on page 300
- "Discovered CITs" on page 301

1 Prerequisites

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** 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

For credentials information, see:

- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

4 Discovery Workflow

- a** In the Discovery Control Panel window, run the **Discovery Modules > Network Discovery > Basic > Range IPs by ICMP** job.
- b** Run **Network Discovery > Basic > Host Connection by Shell**. This job discovers hosts running NTCmd, Telnet, or SSH agents.
- c** Run **Discovery Modules > Network Discovery > Host Resources and Applications > Host Resources and Applications by Shell**. This job discovers running software and processes relevant to WebLogic.

5 Discovered Elements

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 299.

6 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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.
- ▶ 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.

23

WebSphere

This chapter includes:

Concepts

- ▶ WebSphere Discovery Overview on page 304

Tasks

- ▶ Discover WebSphere by JMX on page 305
- ▶ Discover WebSphere by Shell on page 309

Troubleshooting and Limitations on page 312

Concepts

WebSphere Discovery Overview

This section describes how to discover WebSphere application center. 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 305
- "Supported Versions" on page 306
- "Trigger Queries" on page 306
- "Network and Protocols" on page 307
- "Discovery Workflow" on page 307
- "Adapter Parameters for J2EE WebSphere by Shell or JMX" on page 308
- "Discovered CIs" on page 309

1 Prerequisites

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*.

The Probe installation includes WebSphere drivers for versions 5 and 6, but you can use your own drivers, if you prefer. However, you can use only drivers that work with a supported version. For details on supported versions, see "Discovered Applications" on page 24.

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

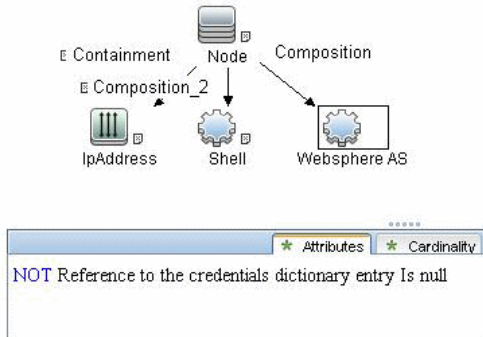
```
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\
discoveryResources\j2ee\websphere\
```

2 Supported Versions

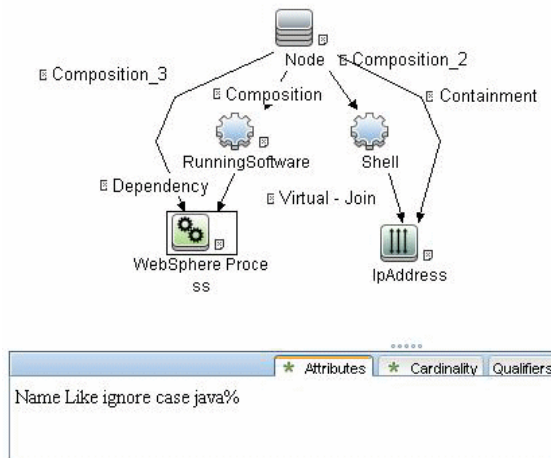
WebSphere versions 5 and 6.

3 Trigger Queries

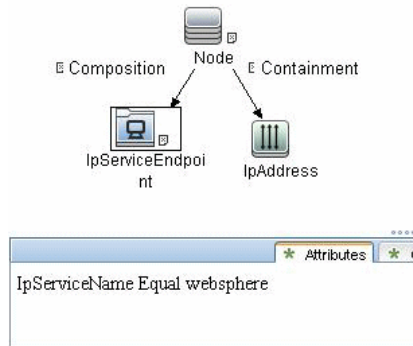
- The J2EE WebSphere by Shell or JMX job:



- The J2EE WebSphere by Shell job:



- The J2EE WebSphere Connections by JMX job:



4 Network and Protocols

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

5 Discovery Workflow

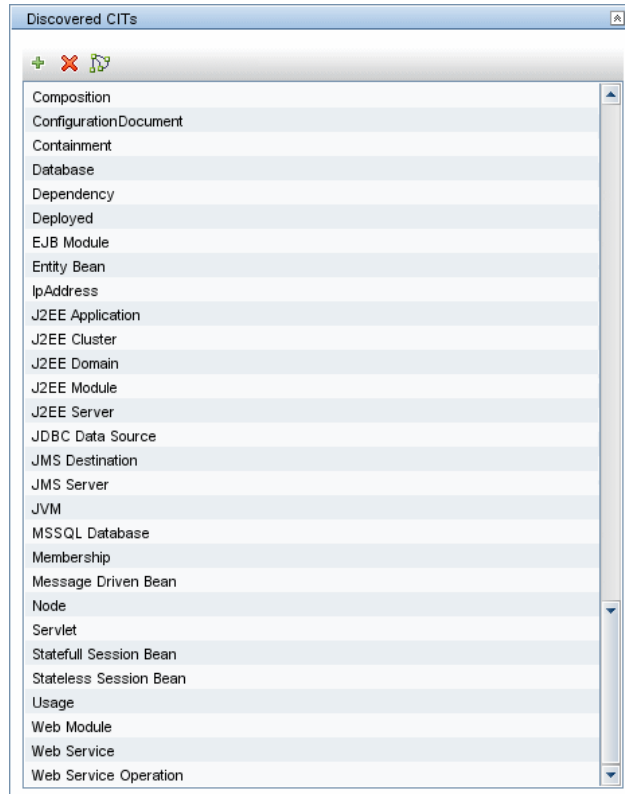
- In the Discovery Control Panel window, run one of the following jobs in the **Discovery Modules > Network Discovery > Basic module**:
 - Range IPs by ICMP
 - Range IPs by NMAP
- Run the **Discovery Modules > J2EE Application Servers > WebSphere > J2EE TCP Ports** job or the **Discovery Modules > Discovery Tools > TCP Ports** job.
- Run the **J2EE WebSphere by JMX** job.
- Run the following jobs:
 - J2EE TCP Ports
 - J2EE WebSphere Connections by JMX
 - J2EE WebSphere by Shell or JMX

6 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.

7 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 310
- "Prerequisites" on page 310

- "Supported Versions" on page 311
- "Network and Protocols" on page 311
- "Discovery Workflow" on page 311
- "Discovered Elements" on page 311
- "Discovered CITs" on page 312

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

- a** Verify that an NTCmd, Telnet, or SSH agent is running on the host.
- b** Verify that the WebSphere server is up and running on the host.
- c** The following procedure is relevant if you are running a client machine that 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*.

3 Supported Versions

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

4 Network and Protocols

For credentials information, see:

- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet 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.

5 Discovery Workflow

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

6 Discovered Elements

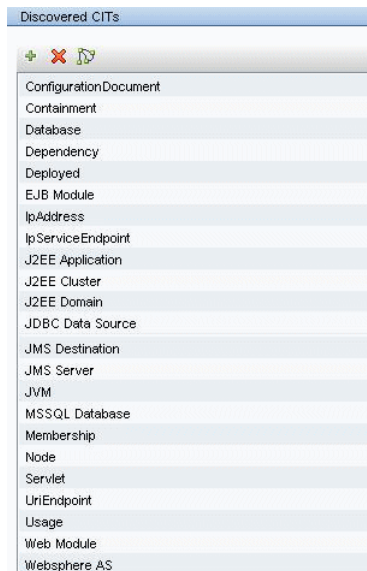
DFM discovers the following elements:

- **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.

7 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 limitations for WebSphere discovery.

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.
- A job (script) works with a certificate in **jks*** key format only.

24

Active and Passive Discovery Network Connections

This chapter includes:

Concepts

- ▶ Overview on page 316

Tasks

- ▶ Discover Processes on page 317
- ▶ Discover TCP Traffic on page 324

Concepts

Overview

All jobs in these modules run queries against the Data Flow Probe's MySQL database to retrieve network connectivity information inserted by the **Host Resources and Applications** and/or **TCP By Shell/SNMP** and/or **Collect Network Data by Netflow** jobs. For details on Host Resource jobs, see "Host Resources and Applications Overview" on page 372.

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 317
- "Job Order and Scheduling" on page 318
- "Supported Versions" on page 318
- "Override Process Parameters (Optional)" on page 318
- "Network and Protocols" on page 318
- "Discovery Workflow" on page 319
- "The IP Traffic by Network Data Job" on page 321
- "The Potential Servers by Network Data Job" on page 322
- "The Server Ports by Network Data Job" on page 321
- "The Servers by Network Data Job" on page 319
- "Discovered CITs" on page 323

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. For details, see "Host Resources and Applications" on page 371.

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 Override Process Parameters (Optional)

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. For details on overriding a process parameter value, see "Parameters Pane" in *HP Universal CMDB Data Flow Management Guide*.

5 Network and Protocols

To discover network connections, define the following protocols:

- "SNMP Protocol"
- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet Protocol"
- "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:

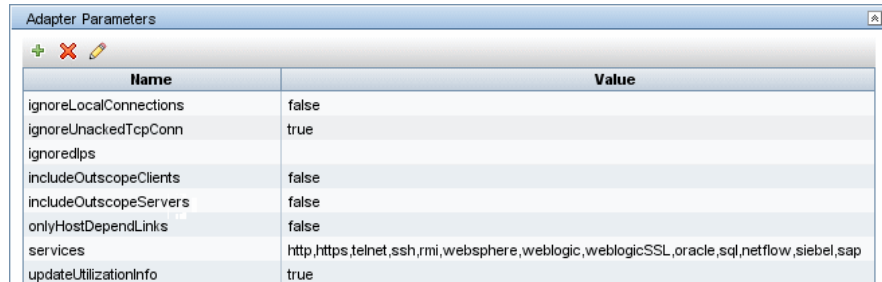
- "The Servers by Network Data Job" on page 319 (**Network Connections > Passive Discovery**)
- "The IP Traffic by Network Data Job" on page 321 (**Network Connections > Active Discovery** and **Network Connections > Passive Discovery**)
- "The Server Ports by Network Data Job" on page 321 (**Network Connections > Active Discovery**)
- "The Potential Servers by Network Data Job" on page 322(**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 adapter parameters:



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

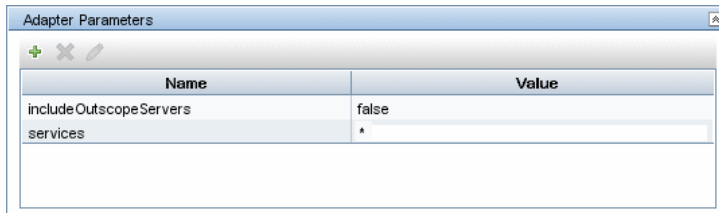
You can configure the maximum number of recognized ports (in which you are interested) through the `maxPorts` parameter.

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 adapter parameters:



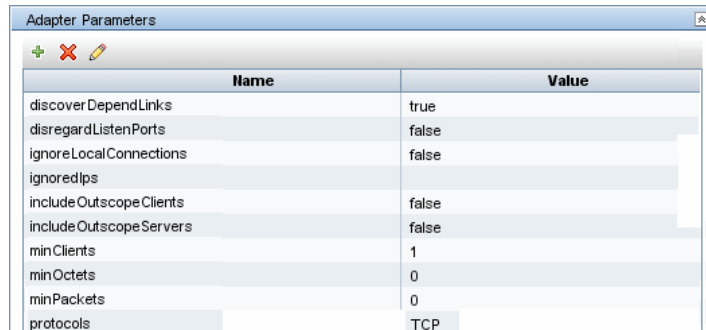
Name	Value
includeOutscopeServers	false
services	*

- **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 adapter parameters:



Name	Value
discoverDependLinks	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 query 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

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

- **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.

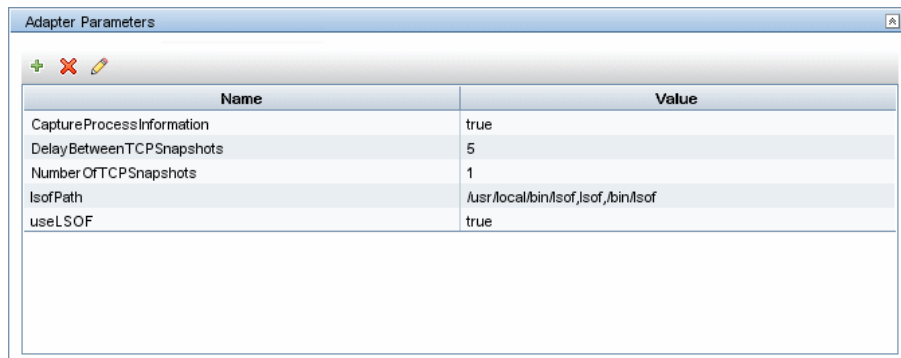
Discover TCP Traffic

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:



Name	Value
CaptureProcessInformation	true
DelayBetweenTCPSnapshots	5
NumberOfTCPSnapshots	1
IsOfPath	/usr/local/bin/IsOf,IsOf,/bin/IsOf
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 "Discover Host Resources and Applications" on page 375.

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 377.

25

Network Basic

This chapter includes:

Concepts

- ▶ Network – Basic Overview on page 328
- ▶ Network Workflow Overview on page 328

Tasks

- ▶ Discover Host Connection by Shell on page 329
- ▶ Discover Host Connection by SNMP on page 332
- ▶ Discover Host Connection by WMI on page 335
- ▶ Discover Windows Running F-Secure with the Host Connection by Shell Job on page 339

Reference

- ▶ Windows Processes on page 340
- ▶ UNIX-Based Processes on page 342

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 Discovery 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 329.
- ▶ **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 "Discover Host Connection by SNMP" on page 332.
- ▶ **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 335.

Tasks

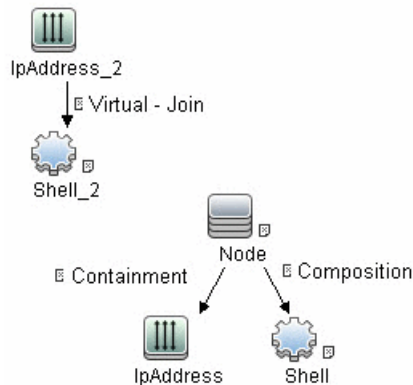
Discover Host Connection by Shell

This subject includes the following sections:

- "Input" on page 329
- "Discovery Workflow" on page 330
- "Discovered CITs" on page 331

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 Protocol"
 - "SSH Protocol"
 - "Telnet Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

2 Discovery Workflow

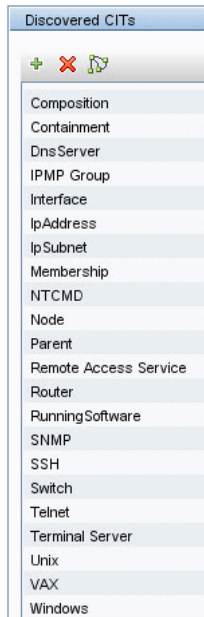
This part of the discovery depends on whether you are discovering components installed on Windows machines or UNIX-based machines. For details on the DFM processes, see:

- "Windows Processes" on page 340.
- "UNIX-Based Processes" on page 342

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



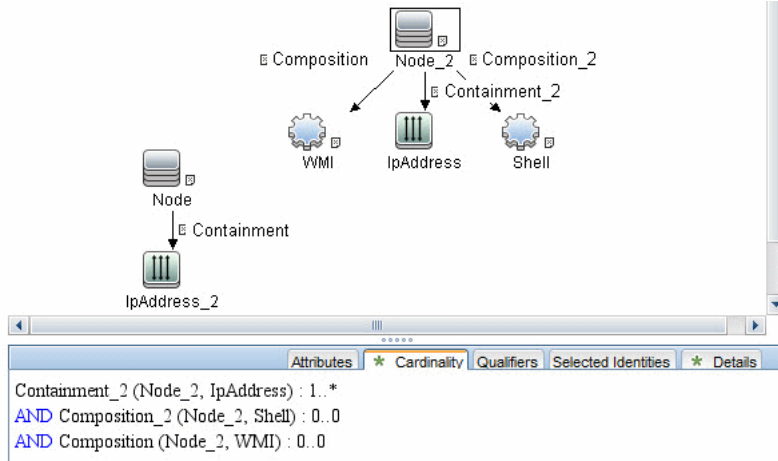
Discover Host Connection by SNMP

This subject includes the following sections:

- "Input" on page 332
- "Discovery Workflow" on page 333
- "Discovered CITs" on page 334

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

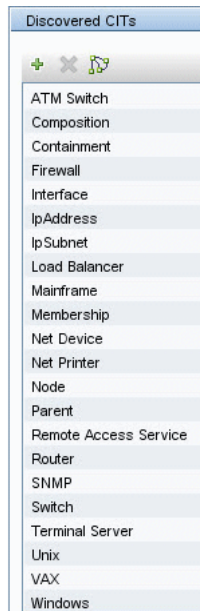
- d** DFM retrieves the network interface information:

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



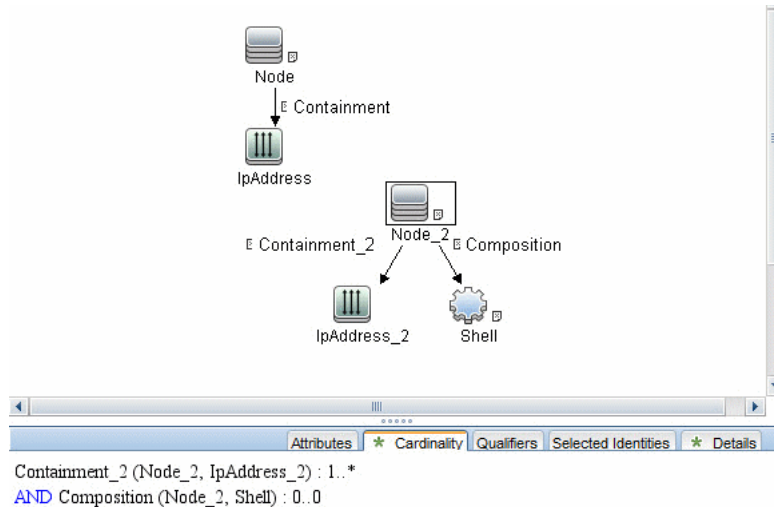
Discover Host Connection by WMI

This subject includes the following sections:

- "Input" on page 335
- "Discovery Workflow" on page 336
- "Discovered CITs" on page 338

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

- a DFM runs through the credentials defined for the WMI protocol and tries to connect successfully through one of them.
- b 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 comply with 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,BuildNumber,Organization,RegisteredUser,TotalVisibleMemorySize,LastBootUpTime,OtherTypeDescription 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
```

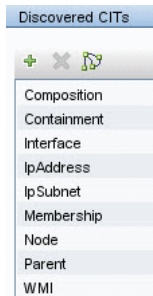
h DFM queries Win32_SystemEnclosure to retrieve the system asset tag.

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



Discover Windows Running F-Secure with the Host Connection by Shell Job

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:

DFM defines the OS. For details, see:

- "FreeBSD" on page 343
- "AIX" on page 344
- "LINUX" on page 345
- "HPUX" on page 346
- "SunOs" on page 346
- "VMKernel" on page 347

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

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'
```

26

Credential-less

This chapter includes:

Concepts

- ▶ Overview on page 350

Tasks

- ▶ Discover Host Fingerprint with Nmap on page 351

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:

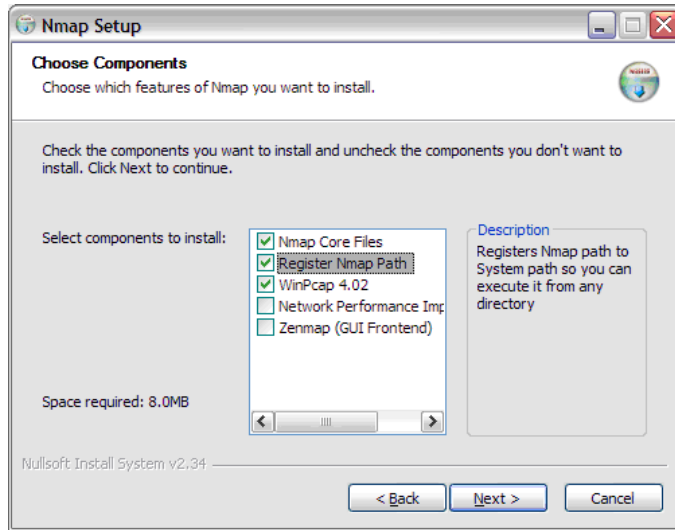
- "Prerequisites" on page 351
- "Discovery Workflow" on page 354
- "Adapter Parameters" on page 355
- "Discovered CITs" on page 355
- "Troubleshooting and Limitations" on page 356

1 Prerequisites

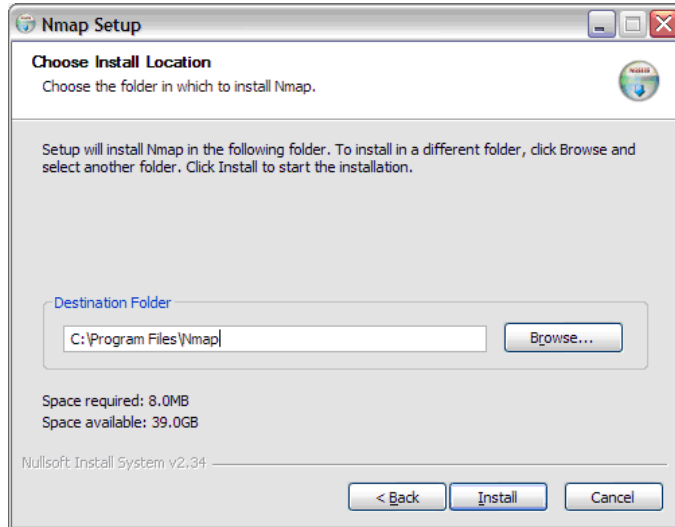
Perform the following procedure on every Data Flow Probe machine that is to run the Host Fingerprint using nmap job:

- 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.

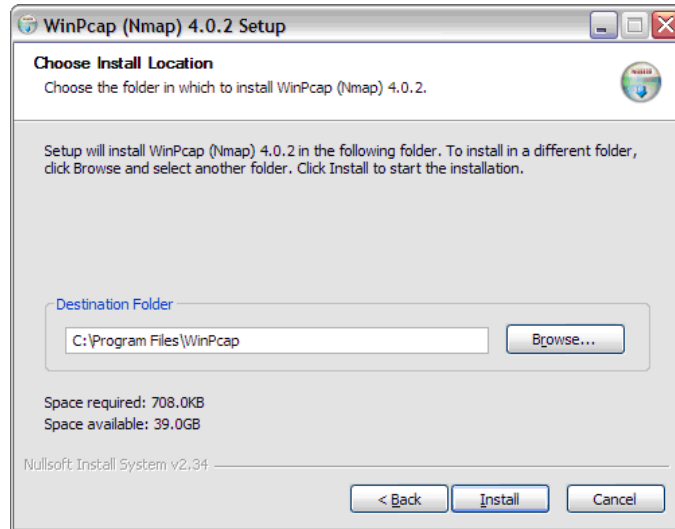


Click **Next**. The **Choose Install Location** dialog box opens.

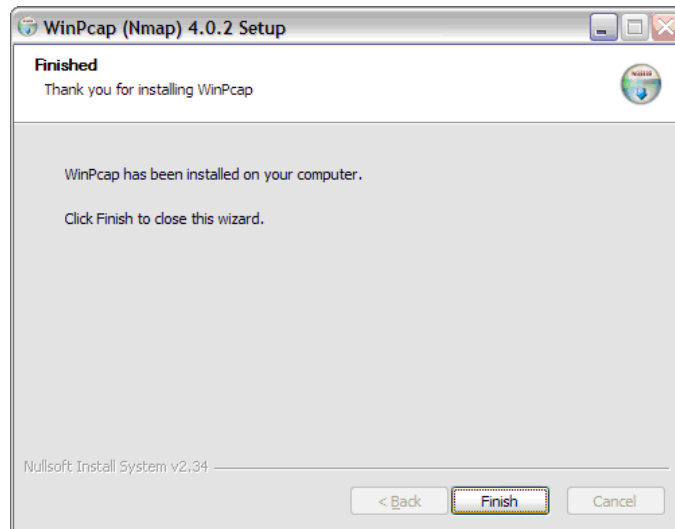


- d Accept the default location or enter another location. Click **Install**.
Nmap is installed. The WinPcap installation dialog box opens immediately after the Nmap installation is complete.

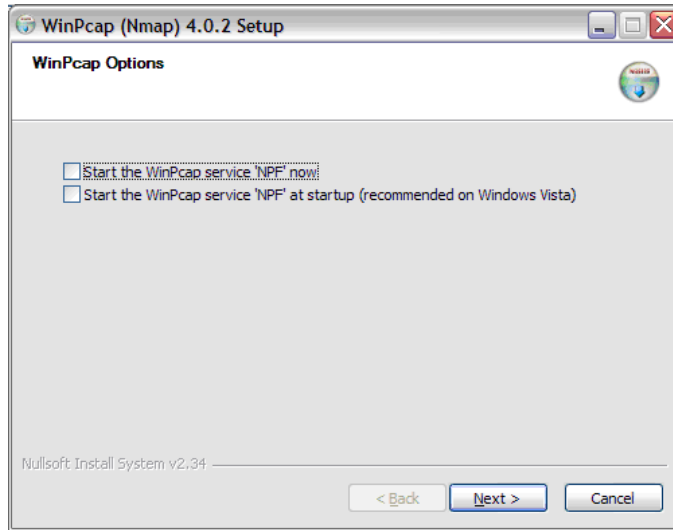
- e Accept the terms of the license and click **Next**. The **Choose Install Location** dialog box opens.



- 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

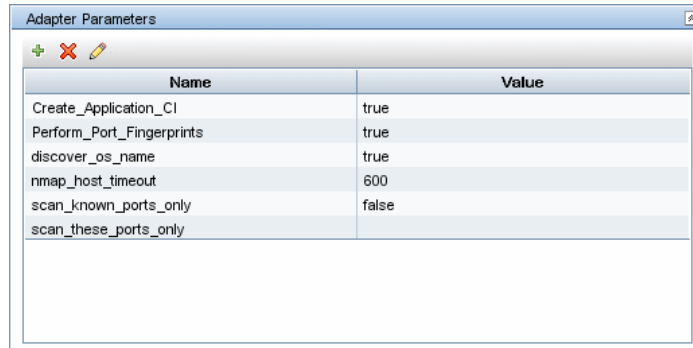
To verify, access the **Add/Remove Programs** window.

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

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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.
The system cannot execute the specified program (in the communication log file)	The Windows system cannot launch the Nmap application.	<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 351.</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>
Nmap is not installed on Probe machine	Nmap is not installed on the Probe machine.	Try to launch Nmap from the command line. Make sure that Nmap is installed. For details on the installation, see "Prerequisites" on page 351.

27

Network – DNS

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 358

Tasks

- ▶ Discover DNS Zone by Nslookup on page 360
- ▶ Discover DNS Zone by DNS on page 363

Reference

- ▶ Discovery Mechanism – Windows on page 366
- ▶ Discovery Mechanism – UNIX-like on page 367
- ▶ Glossary on page 369

Concepts

Overview

DNS Zone discovery retrieves the DNS Zone topology and records that belong to the zone. To transfer the zone, the machine performing the query should be included in a white list configured in the name server. This method requires a special DNS server configuration to permit Probe zone transfer.

The discovery mechanism triggers on a particular name server that records which zones should be reported, as follows:

- 1 Checks the **zoneList** parameter for the list of zones to transfer alias records.
- 2 Ignores zones with the name **arpa**, **localhost**, or **'.'** (root).
- 3 For each zone, transfers all records of type **CNAME** and **A** (second step). If the transfer fails, the zone is not reported.
- 4 Creates realization links.

For details, see "Adapter Parameters" on page 361.

DNS Zone discovery is implemented in the following ways:

- ▶ The **DNS Zone by Nslookup** job queries the DNS server for zone records from the Server itself. This method requires Shell access. For details, see "Discover DNS Zone by Nslookup" on page 360.
- ▶ The **DNS Zone by DNS** job queries the DNS server for zone records from the Data Flow Probe machine. This method requires a special DNS server configuration to permit Probe zone transfer. For details, see "Discover DNS Zone by DNS" on page 363.

In the case where administrators do not want to add Shell access to DNS servers or read access to the configuration file, you can transfer zones specified in the mandatory **zoneList** adapter parameter. For details, see "Adapter Parameters" on page 361.

These implementations retrieve the same topology and have a common discovery mechanism that differs only in the client type (Server or Probe).

Note: The volume of retrieved topology data may be influenced by the parameters set for particular jobs.

Tasks

Discover DNS Zone by Nslookup

This task includes the following steps:

- "Supported Versions" on page 360
- "Prerequisites" on page 360
- "Set up Protocols" on page 361
- "Adapter Parameters" on page 361
- "Trigger Query" on page 362
- "Input Query" on page 362
- "Triggered CI Data" on page 362
- "Discovery Workflow" on page 362
- "Created/Changed Entities" on page 363

1 Supported Versions

- Microsoft Windows 2000 Advanced Server or later
- UNIX-like OS BIND 9 name server

2 Prerequisites

- a If some commands are configured to run with **sudo** on the target host, in the **Protocol Parameters** dialog box, fill in the following fields:
 - **Sudo paths.** Enter the full path to the **sudo** executable, together with the name of the executable. You can add more than one entry if executable files are placed in various places on the target operating systems.

Example: sudo,/usr/bin/sudo,/bin/sudo

- **Sudo commands.** Enter a list of the commands that are prefixed with the **sudo**.

Example: `lspath,ifconfig`

- b** Before activating discovery, confirm that the discovery user has all the required permissions to run the following command:

`cat <path to named config file and its include files>`

For details, see "Protocol Parameter Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

3 Set up Protocols

For credentials information, see:

- "SSH Protocol"
- "NTCMD Protocol"
- "Telnet Protocol"

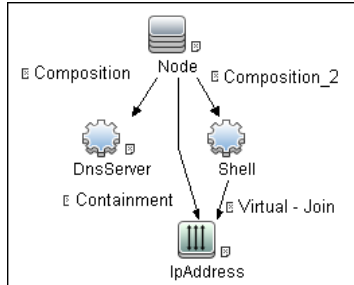
in *HP Universal CMDB Data Flow Management Guide*.

4 Adapter Parameters

The adapter includes the following parameters:

- **isOutOfRangeIpReported.** **False:** The IP is not reported if the IP address is out of the Probe's range. **True:** The IP is reported even if the IP address is out of the Probe's range. The default value is **false**.
- **reportBrokenAliases.** **True:** aliases that do not include a canonical resource are reported. This parameter is needed when an alias points to the address record or another alias record and this record cannot be found in the transferred data. The default value is **false**.
- **zoneList.** A comma-separated list of zones is an optional attribute for the **DNS Zone by Nslookup** job and mandatory for the **DNS Zone by DNS** job. (If it is not set, an error is raised.) The list provides the names of zones that should be transferred. The default value is an empty value.

5 Trigger Query



Shell attributes. NOT Reference to the credentials dictionary entry is null

IP attributes. NOT IP Probe Name is null

6 Input Query



7 Triggered CI Data

- **credentialsId.** Shell credentials
- **ip_address.** Shell IP address

8 Discovery Workflow

- Run the **Range IPs by ICMP** job.
- Run the **Host Connection by Shell** job.
- Run the **Host Resources and Applications by Shell** job.
- Run the **DNS Zone by Nslookup** job.

For details on running jobs, refer to the "Discovery Control Panel" chapter in *HP Universal CMDB Data Flow Management Guide*.

9 Created/Changed Entities

- The DNS_Zone adapter parameters. For details on the new adapter parameters, see "Overview" on page 358.
- The DNS Zone by Nslookup job
- The DNS Record class (new)

Discover DNS Zone by DNS

This task includes the following steps:

- "Supported Versions" on page 363
- "Prerequisites" on page 363
- "Trigger Query" on page 364
- "Input Query" on page 364
- "Discovery Workflow" on page 364
- "Created/Changed Entities" on page 364

1 Supported Versions

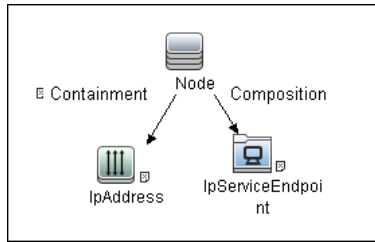
- Microsoft Windows 2000 Advanced Server or later
- UNIX-like OS BIND 9 name server

2 Prerequisites

Discovery is performed by the DNS protocol. To perform discovery, set up the following:

- As all requests are performed from the Probe machine, this machine must be included in the list of servers that can transfer specified zone records. The administrator of the name server grants permissions to transfer the zone from the Probe machine.
- Provide a list of zones that need to be transferred. For details, see "Adapter Parameters" on page 361.

3 Trigger Query



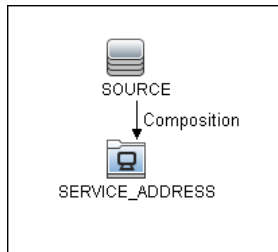
where the IpServiceEndpoint attribute is:

Name Equal dns

AND NOT IP address is null

4 Input Query

Triggered CI Data. ip_address. Shell IP address



5 Discovery Workflow

- a Run the **Range IPs by ICMP** job.
- b Run the **TCP ports** job.
- c Run the **DNS Zone by DNS** job.

6 Created/Changed Entities

- The **DNS_Zone_By_Shell** adapter parameters.
- The **DNS Zone by Shell** job
- The **Network – DNS** module

- ▶ The dns_service Trigger query
- ▶ The DNS Record class (new)

Reference

Discovery Mechanism – Windows

This section includes the following commands:

- ▶ "Query Windows Registry for Zone Information" on page 366
- ▶ "List Root Domain to Transfer Resource Records" on page 367

Query Windows Registry for Zone Information

Command

Reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS Server\Zones"

Output

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS
Server\Zones\104.24.172.in-addr.arpa
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS
Server\Zones\foo.bar.net
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS
Server\Zones\od5.lohika.com
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS
Server\Zones\ucmdb-ex.dot
```

Mapping

CMD Output Attribute	CI Name	CI Attribute
Key name	DNS Zone	Name

List Root Domain to Transfer Resource Records

Zone resource records of type **CNAME** and **A** are transferred by listing the root domain of the zone in the **nslookup** command.

Command

```
echo ls -d <domain> | nslookup - <name server>
```

Output

```
Ns-2.od5.lohika.com.      CNAME  dc05-2.od5.lohika.com
od5.lohika.com.          A      134.44.98.22
ftp.od5.lohika.com.     CNAME  od5.lohika.com.
```

Mapping

CMD Output Attribute	CI Name	CI Attribute
First column	DNS Alias	Name
Third column	DNS Alias	Canonical name

Discovery Mechanism – UNIX-like

This section includes the following commands:

- "Parse Named Server Configuration File to Retrieve Zone Information" on page 367
- "List Root Domain to Transfer Resource Records" on page 368

Parse Named Server Configuration File to Retrieve Zone Information

- 1 Try to find information about the named server configuration file in the command like the corresponding process.

Command

```
ps -ef | grep named | awk '{for(i=11; i < NF; i++) {printf("%s ", $i)}printf("\n")}'
```

Output

```
/usr/sbin/named -t /var/lib/named -u
```

Mapping

The path specified for the `-t` option is the path to the configuration file.

- 2 If the path is recognized, the job tries to retrieve information about zones and include files to process. The default paths are `/etc/named.conf` and `/etc/namedb/named.conf`.

Command

```
cat <configuration file path> | awk '/zone|include/{print}'
```

Output

```
zone "." in {
zone "localhost" in {
zone "od5.lohika.com" in {
```

Mapping

CMD Output Attribute	CI Name	CI Attribute Display Name
Key name	DNS Zone	Name

List Root Domain to Transfer Resource Records

Zone resource records of type **CNAME** and **A** are transferred using the **dig** command and the **axfr** transfer type.

Command

```
dig @<server> <domain> axfr | awk '/(CNAME|A)/{print $1, "\t", $4, "\t", $5}'
```

Output

```
Ns-2.od5.lohika.com.      CNAME  dc05-2.od5.lohika.com
od5.lohika.com.          A      134.44.98.22
ftp.od5.lohika.com.      CNAME  od5.lohika.com.
```


Mapping

CMD Output Attribute	CI Name	CI Attribute Display Name
First column	DNS Alias	Name
Third column	DNS Alias	Canonical name

Glossary

➤ **CNAME record or Canonical Name record**

A type of resource record in the Domain Name System (DNS) that specifies that the domain name is an alias of another canonical domain name.

➤ **Zone transfer**

Listings of records contained in the zone.

28

Host Resources and Applications

This chapter includes:

Concepts

- ▶ Host Resources and Applications Overview on page 372

Tasks

- ▶ Discover Host Resources and Applications on page 375
- ▶ Revert to Previous Method of Discovering Installed Software on page 381

Troubleshooting and Limitations on page 381

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.

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).

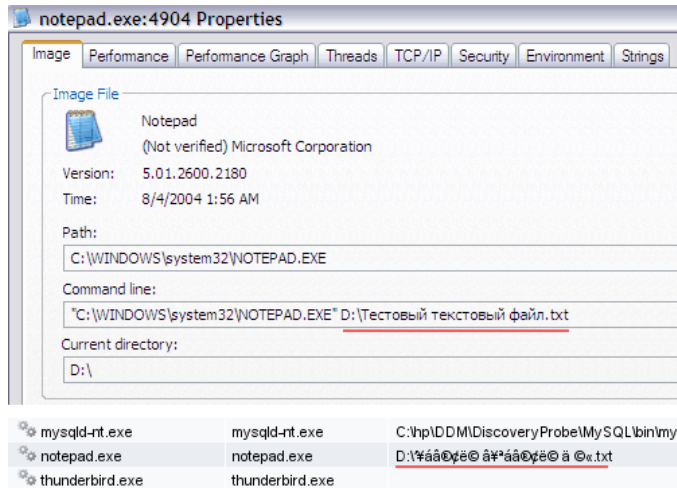
Note:

- ▶ 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

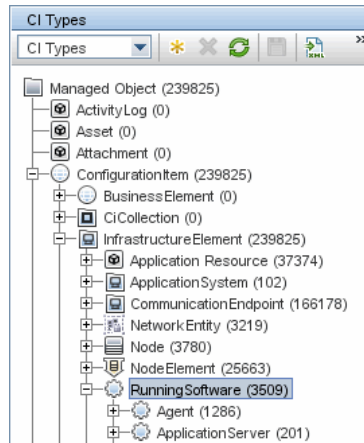
Discover Host Resources and Applications

This task includes the following steps:

- "Prerequisites" on page 376
- "Network and Protocols" on page 376
- "Adapter Parameters for the Host Resources and Applications by Shell Job" on page 377
- "Adapter Parameters for the Host Resources and Applications by SNMP Job" on page 378
- "Adapter Parameters for the Host Resources and Applications by WMI Job" on page 379
- "Discovery Workflow for the Host Resources and Applications by Shell/SNMP/WMI Jobs" on page 379
- "Discovery Workflow for the Software Element CF by Shell Job" on page 379
- "TCP Discovery" on page 380
- "Discovered CITs" on page 380
- "Topology Map" on page 380

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:

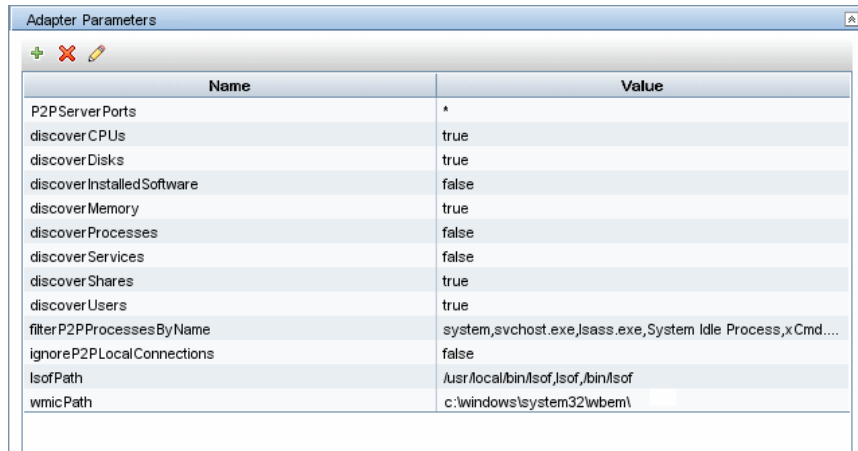
- "NTCMD Protocol"
- "SNMP Protocol"
- "SSH Protocol"
- "Telnet Protocol"
- "WMI Protocol"

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*.



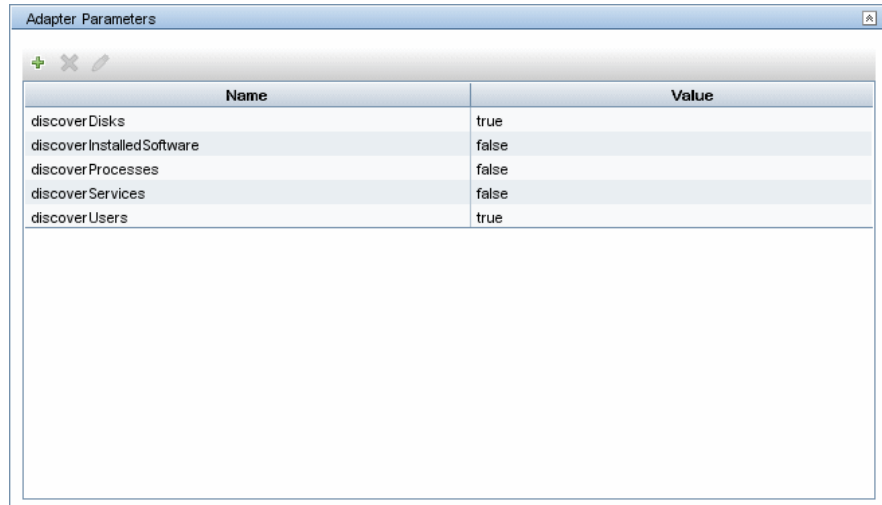
Name	Value
P2P Server Ports	*
discover CPUs	true
discover Disks	true
discover Installed Software	false
discover Memory	true
discover Processes	false
discover Services	false
discover Shares	true
discover Users	true
filter P2P Processes By Name	system,svchost.exe,lsass.exe,System Idle Process,xCmd....
ignore P2P Local Connections	false
lsOfPath	Ausr/local/bin/lsOf,lsOf,/bin/lsOf
wmicPath	c:\windows\system32\wbem\

- **P2P Server Ports.** 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 377.

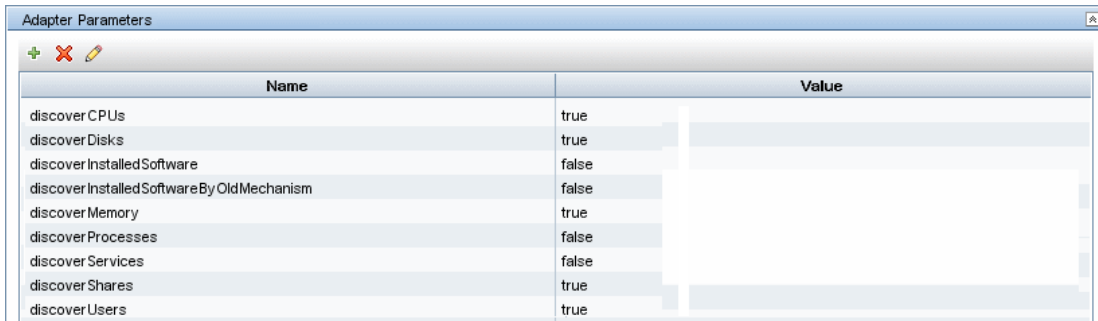


The screenshot shows a window titled "Adapter Parameters" with a table containing five rows of data. The table has two columns: "Name" and "Value".

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 377.



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 node (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):

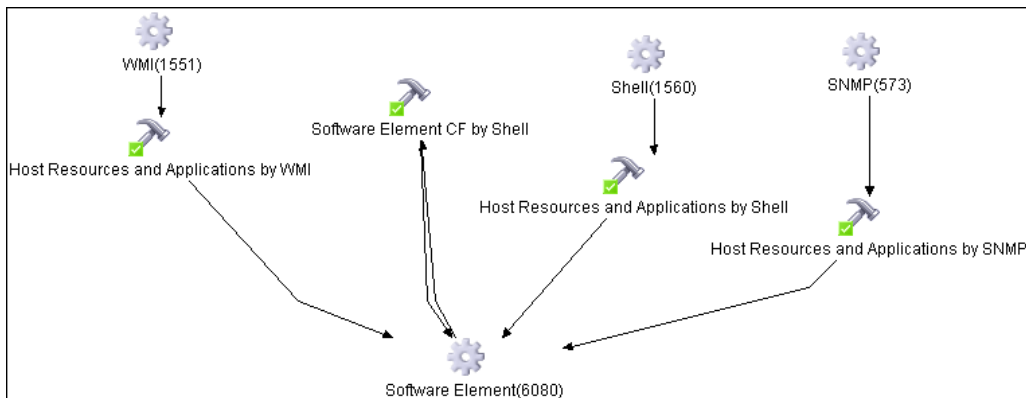
- ▶ If one of the ports is a listening port (that is, is marked as listening in the port_process table), then this port is a server port.
- ▶ If one of the ports is used by a process that is known to be a server process, then this port is the server port.
- ▶ If a local port is not listening and the 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

To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

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.

29

Layer 2

This chapter includes:

Concepts

- ▶ Overview on page 384

Tasks

- ▶ Discover Layer 2 Objects on page 385

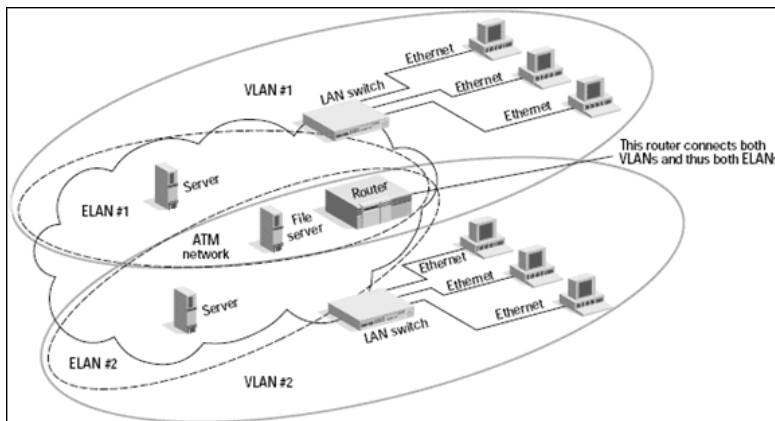
Concepts

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 CIs 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 386
- "Discovery Workflow" on page 387
- "Discovered CITs: VLANs by SNMP" on page 390
- "Discovered CITs: VLAN ports by SNMP" on page 390
- "Discovered CITs: Layer2 Topology Bridge Based by SNMP" on page 390
- "Discovered CITs: Layer2 Topology VLAN Based by SNMP" on page 391
- "Layer 2 Relationships" on page 391
- "Troubleshooting and Limitations" on page 392

1 Prerequisites

Caution:

- ▶ All network connection jobs should finish running before you activate the Layer 2 jobs (consisting of the following jobs: Host Networking by SNMP, Layer2 Topology Bridge based by SNMP, Layer2 Topology VLAN based by SNMP, VLAN Ports by SNMP, VLANs by SNMP).
 - ▶ 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. (Layer2 CIs names are the same as the relevant interface name and interface description on the destination network interface adapter which we are discovering.
-

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 Host Networking by SNMP job for the Layer 2 module.
-

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** query. 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%**

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** query. 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*.

The `SNMP_Net_Dis_VMS_catalyst.py` script retrieves the Base MAC table and Port number If Index table.

c Activate the **Layer2 Topology Bridge based by SNMP** job.

The trigger for this job is the `catalyst_bridge_no_vlan` query. 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%**

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 on page 387, 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** query. 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%**

For details on the `bridgePortDisc.py` script, see step c on page 388.

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.

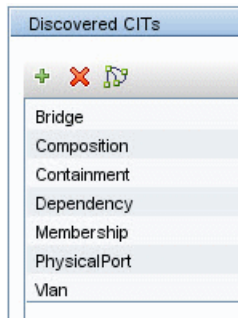
- f Activate the **Host Networking by SNMP** job.

This job discovers host networking topology using SNMP route and system tables.

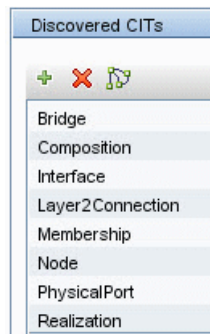
3 Discovered CITs: VLANs by SNMP

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

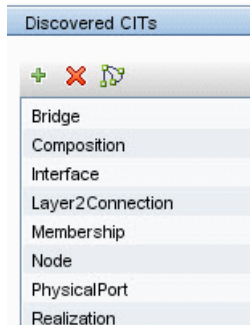
4 Discovered CITs: VLAN ports by SNMP



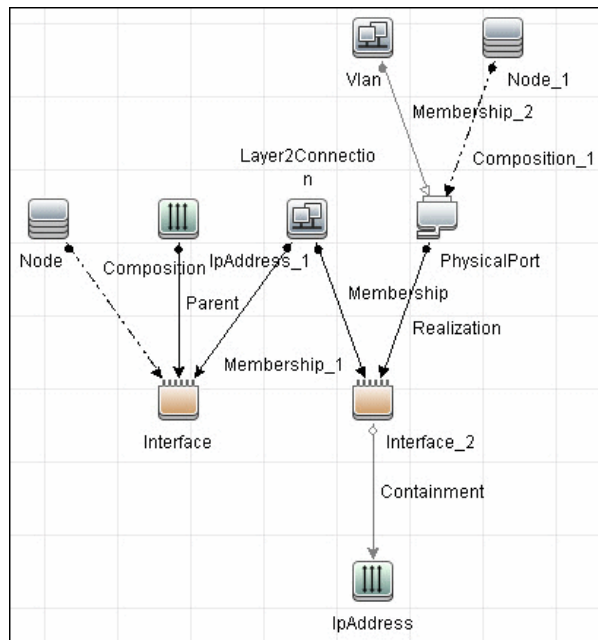
5 Discovered CITs: Layer2 Topology Bridge Based by SNMP



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 port on the switch can be connected to a host or interface object (the end user machines) by a Layer 2 CI, or to a port-switch by a Backbone link.

8 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`.

30

Discovery Tools

This chapter includes:

Concepts

► Overview on page 400

Troubleshooting and Limitations on page 400

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 32, "Importing Data from External Sources."

Troubleshooting and Limitations

This section describes troubleshooting and limitations for Siebel discovery, when running the **File Monitor by Shell** job.

Problem: The **File Monitor by Shell** does not trigger automatically. This is because there is no trigger query 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.

31

Import from Excel Workbook

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Tasks

► Discover Import from Excel Workbook on page 402

Troubleshooting and Limitations on page 405

Tasks

Discover Import from Excel Workbook

This doc describes the usage and functionality of the **XLS_Import** package. The job imports data from the Probe's file system (or accessible network share), so no credentials are required.

This task includes the following steps:

- "Supported Versions" on page 402
- "Prerequisites" on page 402
- "Trigger Query/Input Query" on page 404
- "Discovery Workflow" on page 404

1 Supported Versions

The XLS_Import package supports Microsoft Excel files, versions 97, 2000, XP, and 2003 (*.xls) as well as Office Open XML format for Excel 2007 (*.xlsx).

2 Prerequisites

- a** Set up permissions:

Give the Data Flow Probe read permissions to the location on the file system where the import files are stored.

b Modify the Probe class path:

- Edit the following file:
C:\hp\UCMDB\DataFlowProbe\bin\WrapperEnv.conf
- Locate the **Environment global vars** section and add following line to the end of the section:

```
set.probeManager=%runtime%/probeManager
```

- Locate the **Environment Discovery Path** section and add the following line:

```
set.POI_CLASSES=%probemanager%/discoveryResources/geronimo-stax-api_1.0_spec-1.0.jar;%probemanager%/discoveryResources/poi-3.7-beta1-20100620.jar;%probemanager%/discoveryResources/poi-ooxml-3.7-beta1-20100620.jar;%probemanager%/discoveryResources/poi-ooxml-schemas-3.7-beta1-20100620.jar;%probemanager%/discoveryResources/xmlbeans-2.3.0.jar
```

- Use either of the following steps, according to your environment:
Add the **%POI_CLASSES%** reference before the **%NNM_CLASSES%** reference by appending it to line ~54, for example:

```
set.COMMON_CLASSPATH=%POI_CLASSES%;%conf%;%XML_CLASSES%;%JYTHON_CLASSES%;%NNM_CLASSES%;...
```

or

Add the following line directly after **set.COMMON_CLASSPATH=.....**:

```
set.COMMON_CLASSPATH=%POI_CLASSES%;%COMMON_CLASSPATH%
```

- Restart the Probe.
- c** Verify that the CITs exist:

Each tab in the Excel file is mapped to a specific CI type. Verify that the CIT is defined in the CMDB data model prior to running the job. If you are importing out-of-the-box CITs, you do not have to create the CIT since they already exist in the CMDB.

- d** Verify that the CIT attributes are already defined in the CMDB. If they do not exist, the data is rejected.

3 Trigger Query/Input Query

The **Import from Excel Workbook** job has no trigger query. Therefore, you must manually add the Probe that imports the data. For details, see "Probe Selection Pane" in *HP Universal CMDB Data Flow Management Guide*.

Because the job's input CI type is **Discovery Probe Gateway**, there is no need to supply an input query.

4 Discovery Workflow

The data type of the attribute (string, long, integer, Boolean, and so on) depends on the CMDB data model. You do not need to set attribute types manually. You do have to specify the attribute name in the document header line.

Discovery performs the following validations:

- a** Verifies that the CI Types on the tabs in the spreadsheet exist in the CMDB.
- b** Verifies that the attributes (the column names in the spreadsheet) exist in the CMDB.
- c** Checks the presence of key attributes on the spreadsheet.
- d** Processes all CI Types that contain a **root_container** attribute after CI Types that do not have this type of attribute. This helps to ensure that the parent CI is created before a contained CI.
- e** Processes the **relationships** tab last, to create relationships between CIs that do not use the **Composition** relationship.

For the relationship to be created, the keyed attributes of a CI must be used in the **relationships** tab.

For example, for **node**, you must use **host_key** when creating a containment relationship between **node** and **IpAddress**. In this case, the tab entries would look like following (the quotes show long strings):

host:

host_key	name
'192.168.100.100 MyDomain'	testhost

IpAddress:

ip_address	ip_domain
192.168.100.100	MyDomain

relationships:

start	relation_type	end
'192.168.100.100 MyDomain'	containment	192.168.100.100

Note: The space in the **host_key** value (between **192.168.100.100** and **MyDomain**) is needed for incomplete hosts.

Troubleshooting and Limitations

Problem: Job compile time errors and problems working with the Excel files.

Solution: Verify that you have completed the prerequisites section. For details, see "Prerequisites" on page 402.

32

Importing Data from External Sources

This chapter includes:

Concepts

- ▶ Importing Data from External Sources Overview on page 408
- ▶ The External_source_import Package on page 410
- ▶ The Import from CSV File Job on page 411
- ▶ The Import from Database Job on page 415
- ▶ The Import from Properties File Job on page 419
- ▶ The External Source Mapping Files on page 421
- ▶ Convert Strings to Numbers on page 422

Tasks

- ▶ Import CSV Data from an External Source – Scenario on page 424

Troubleshooting and Limitations on page 428

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:

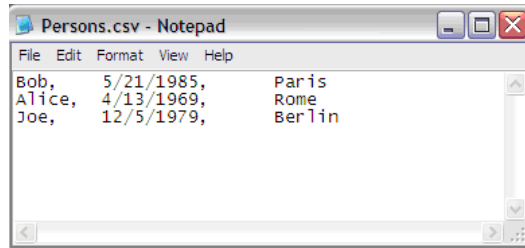
- "Comma Separated Value (CSV) Files" on page 408
- "Databases" on page 409
- "Properties Files" on page 410

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**.

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.

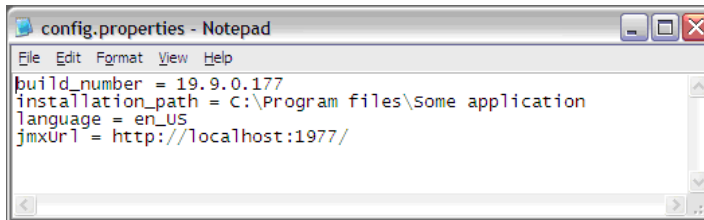
The following databases are supported: Oracle, Microsoft SQL Server, MySQL, and DB2.

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:

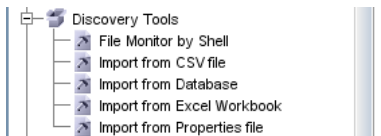


The External_source_import Package

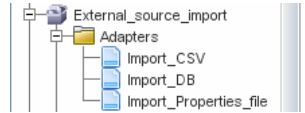
The External_source_import package consists of three jobs and three adapters. There is one job and one adapter for each external source (CSV file, properties file, database):

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:



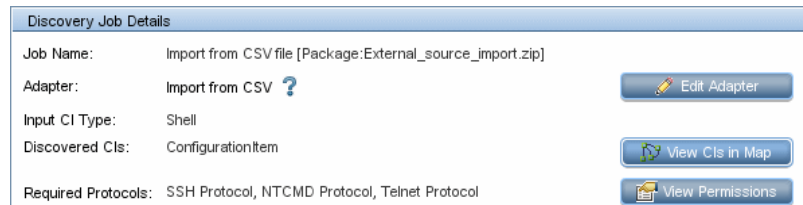
The Import from CSV File Job

This section includes the following topics:

- "Job Details" on page 411
- "Adapter Parameters" on page 412
- "Delimiters, Quotes, and Escaping Characters" on page 413

Job Details

The job details are as follows:



This job has no Trigger queries 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 428.

The Import from CSV File job is located under the Discovery Tools module.

Adapter 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 413.
- ▶ **mappingFile.** For details of the mapping file, see "The External Source Mapping Files" on page 421.
- ▶ **rowToStartIndex.** For details on setting the row at which DFM starts collecting data, see "CSV Files with Column Titles in First Row" on page 409.
- ▶ **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:name** 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 **name** 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 421.
- 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 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 415
- "Discovery Adapter Parameters" on page 415
- "Tables and Queries" on page 416
- "Database, Schema, and Table Names" on page 417
- "Importing Data with a SQL Query" on page 417
- "Column Types" on page 418

Job Details

The job details are as follows:

Discovery Job Details	
Job Name:	Import from Database [Package:External_source_import.zip]
Adapter:	Import from DB ? Edit Adapter
Input CI Type:	Database
Discovered CIs:	ConfigurationItem View CIs in Map
Required Protocols:	SQL Protocol View Permissions

This job has no Trigger queries associated with it.

Discovery Adapter Parameters

The following parameters are included by default:

- **ciType**. For details, see "Adapter Parameters" on page 412.
- **mappingFile**. For details of the mapping file, see "Adapter Parameters" on page 412.
- **mappingString**. For details, see "Adapter Parameters" on page 412.
- **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	
ciType	
mappingFile	
mappingString	
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:

Adapter Parameters	
+ ✖ ✎	
Name	
ciType	
mappingFile	
mappingString	
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 417.

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 UCMDB. Use the `binaryIntToBoolean` converter, as follows:

```
<map>
  <attribute>cluster_is_active</attribute>
  <column type="int">cluster_is_active</column>
  <converter module="import_converters">binaryIntToBoolean</converter>
</map>
```

`type="int"`. This attribute specifies that the value of `cluster_is_active` should be retrieved as an integer, and that the value passed to the converter method should be an integer.

If the `cluster_is_active` attribute of the CIT is of type `integer`, the converter is not needed here, and the mapping file should say:

```
<map>
  <attribute>cluster_is_active</attribute>
  <column type="int">cluster_is_active</column>
</map>
```

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 420
- "Discovery Adapter Parameters" on page 420
- "Keys and Values" on page 420
- "Comments in Properties Files" on page 420

Job Details

The job details are as follows:

Discovery Job Details	
Job Name:	Import from Properties file [Package:External_source_import.zip]
Adapter:	Import from properties file ? Edit Adapter
Input CI Type:	Shell
Discovered CIs:	ConfigurationItem View CIs in Map
Required Protocols:	SSH Protocol, NTCMD Protocol, Telnet Protocol View Permissions

This job has no Trigger queries associated with it.

Discovery Adapter Parameters

The following parameters are included by default:

- ▶ **ciType**. For details, see "Adapter Parameters" on page 412.
- ▶ **mappingFile**. For details of the mapping file, see "Adapter Parameters" on page 412.
- ▶ **mappingString**. For details, see "Adapter Parameters" on page 412.
- ▶ **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 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.

Convert Strings to Numbers

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 424
- "Create a CIT" on page 425
- "Create a Mapping File" on page 426
- "Activate the Import from CSV File Job" on page 427
- "Add the Discovered Shell CI to the Job" on page 428
- "Result" on page 428

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:

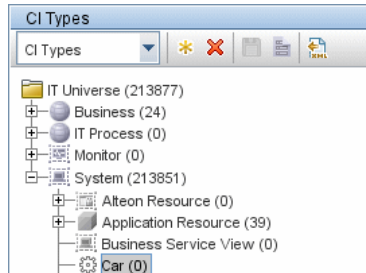
1	Name	string
2	Model	string
3	Year of manufacture	integer
4	Date of purchase	date

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 the *HP Universal CMDB Modeling Guide*.

- b** During the creation of the CIT, the admin adds these attributes as follows:

Key	Name	Display Name	Type
BODY_ICON	BODY_ICON	BODY_ICON	string
root_candidatefordel...	Candidate For Deleti...	Candidate For Deleti...	date
date_of_purchase	Car Date of Purchase	Car Date of Purchase	date
model	Car Model	Car Model	string
name	Car Name	Car Name	string
year_of_manufacture	Car Year of Manufa...	Car Year of Manufa...	integer

For details, see "Attributes Page" in the *HP Universal CMDB 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 "Convert Strings to Numbers" on page 422.

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.**

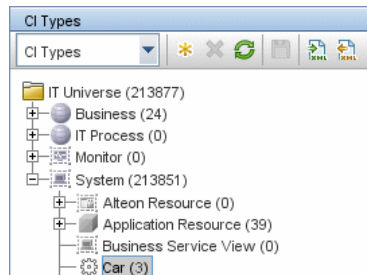
For details on activating jobs, see "Discovery Modules Pane" in *HP Universal CMDB Data Flow Management Guide*.

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 428
- ▶ "Timeout Issues When Importing from CSV and Properties Files" on page 429

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 409.

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.

33

HP Partitioning Solution

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 432

Tasks

- ▶ Discover HP vPars and nPars on page 433

Reference

- ▶ Views on page 437
- ▶ Discovery Mechanism on page 440

Troubleshooting and Limitations on page 469

Concepts

Overview

HP nPartitions

Cell-based HP servers enable you to configure a single server complex as one large system or as multiple smaller systems by configuring **nPartitions**. Each nPartition defines a subset of server hardware resources to be used as an independent system environment. An nPartition includes one or more cells assigned to it (with processors and memory) and all I/O chassis connected to those cells. All processors, memory, and I/O in an nPartition are used exclusively by software running in the nPartition. Thus, each nPartition has its own system boot interface, and each nPartition boots and reboots independently. Each nPartition provides both hardware and software isolation, so that hardware or software faults in one nPartition do not affect other nPartitions within the same server complex. You can reconfigure nPartition definitions for a server without physically modifying the server hardware configuration by using the HP software-based nPartition management tools.

HP vPartitions

vPars is a Virtual Partitions product that enables you to run multiple instances of HP-UX simultaneously on one hard partition by dividing that hard partition further into virtual partitions. Each virtual partition is assigned its own subset of hardware, runs a separate instance of HP-UX, and hosts its own set of applications. Because each instance of HP-UX is isolated from all other instances, vPars provides application and Operating System (OS) fault isolation. Each instance of HP-UX can have different patches and a different kernel.

Tasks

Discover HP vPars and nPars

This task includes the following steps:

- "Supported Version" on page 433
- "Prerequisites" on page 433
- "Deploy the Package" on page 434
- "Trigger Query for the HP nPartitions by Shell Job" on page 434
- "The Input Query for the hp_npar_by_shell Adapter" on page 434
- "Discovery Workflow" on page 435
- "Sample Output" on page 435
- "Created/Changed Entities" on page 435

1 Supported Version

This discovery is relevant for the vPars A.03.xx, A.04.xx, and A.05.xx versions.

This package has been verified on cellular systems with vPars running a HP-UX operating system. Non-cellular systems and vPars running other operating systems are not supported in this version.

2 Prerequisites

- Confirm that Shell credentials are set up on the Probe. For details, see "Domain Credential References" in *HP Universal CMDB Data Flow Management Guide*.

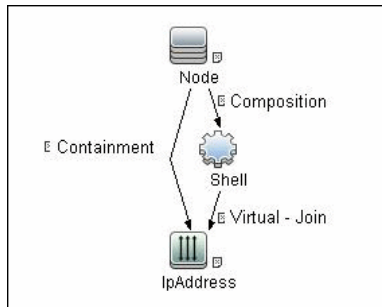
3 Deploy the Package

The name of the package is **HP_nPartitions**.

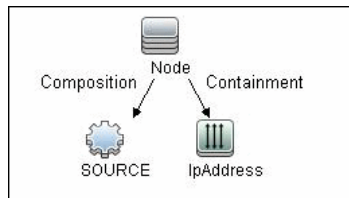
For details on deploying packages, see the "Package Manager" chapter in the *HP Universal CMDB Administration Guide*.

4 Trigger Query for the HP nPartitions by Shell Job

Note: The `host_shell` name is also used by the **Host Resources and Applications by Shell** job.



5 The Input Query for the hp_npar_by_shell Adapter

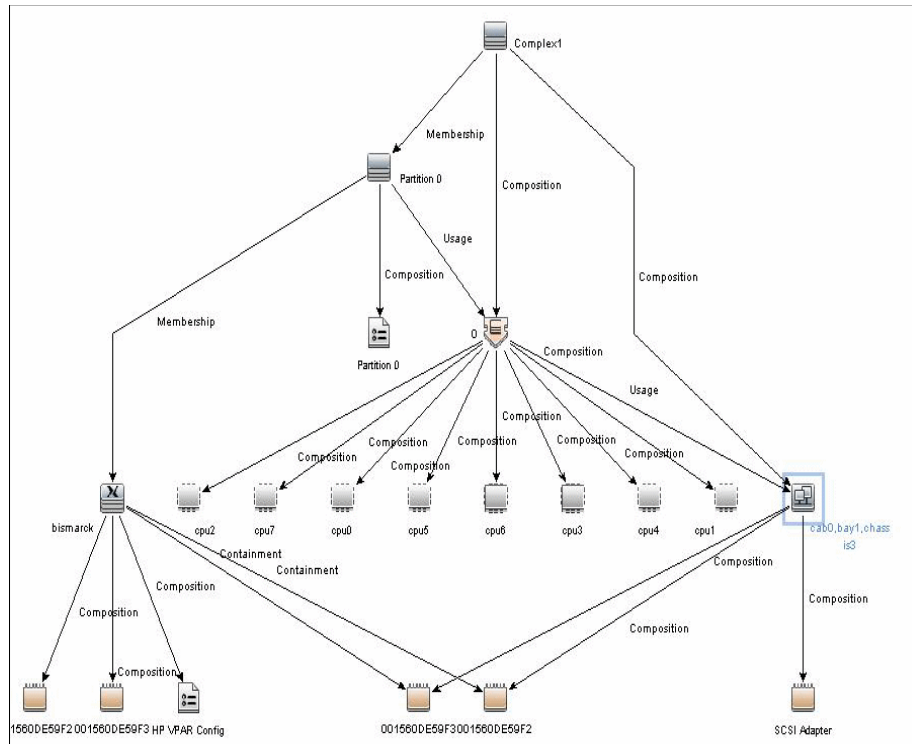


6 Discovery Workflow

For details on jobs, see "Discovery Control Panel – Advanced Mode Workflow" in *HP Universal CMDB Data Flow Management Guide*.

- a Run the **Range IPs by ICMP** job.
- b Run the **Host Connection by Shell** job.
- c Run the **HP nPartitions by Shell** job.

7 Sample Output



8 Created/Changed Entities

New Classes

- hp_complex
- cell_board

- io_chassis
- hp_npar_config
- hp_vpar_config

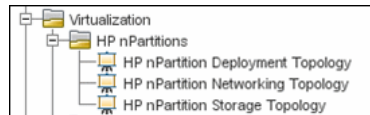
New Links

End1	Link Type	End2
node	containment	fchba
node	containment	interface
node	containment	scsi_adapter
cell_board	composition	cpu
cell_board	composition	memory
hp_complex	composition	io_chassis
io_chassis	composition	fchba
io_chassis	composition	interface
io_chassis	composition	scsi_adapter
cell_board	usage	io_chassis
node	usage	cell_board
node	usage	fchba
node	usage	interface

Reference

Views

HP nPartitions topology is represented by the following views under the Virtualization module:

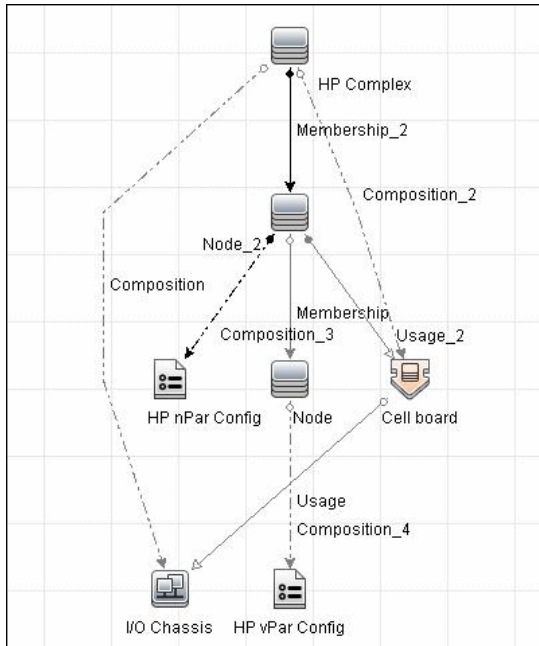


This section includes the following topics:

- "HP nPartition Deployment Topology View" on page 438
- "HP nPartition Networking Topology View" on page 439
- "HP nPartition Storage Topology View" on page 440

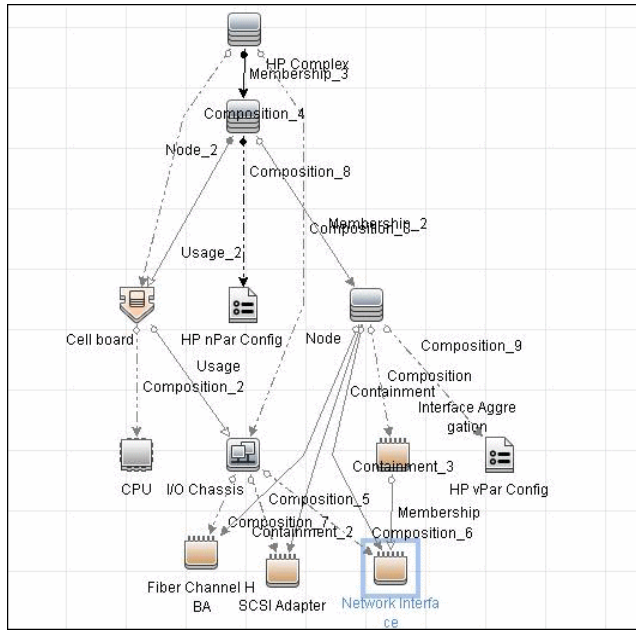
HP nPartition Deployment Topology View

This view represents the basic virtualization deployment, containing nPars, vPars, cells, and I/O chassis only.



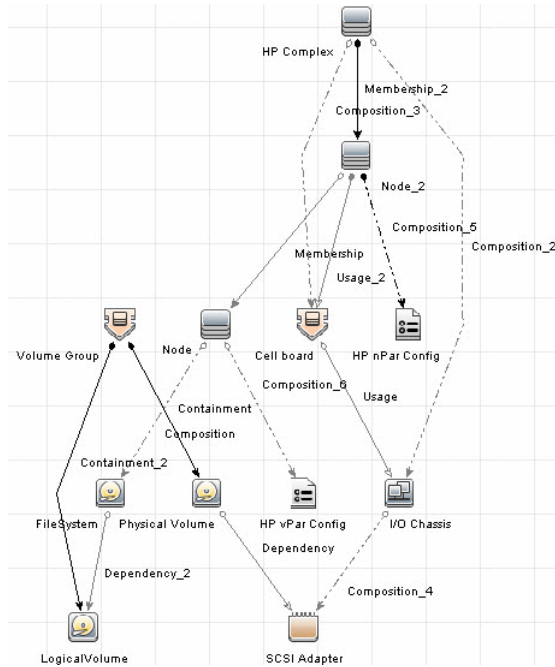
HP nPartition Networking Topology View

This view represents the Networking aspect of the nPartition deployment including the relations between I/O devices of vPars and their physical locations on the I/O chassis.



HP nPartition Storage Topology View

This view reflects the storage aspect of the HP nPartitions system including the relations between file systems and logical volumes.



Discovery Mechanism

This section includes the following commands:

- "Verify Discovery on the vPartition" on page 441
- "Verify Discovery on the nPartition" on page 442
- "Get Information about Complex" on page 442
- "List General Information About All Cells" on page 444
- "List Detailed Information About Each Cell" on page 444
- "Get Information About I/O Chassis" on page 450
- "Get the List of Names of the nPartitions on the System" on page 451

- "Get Detailed Information About nPartition" on page 451
- "Get the Name of the Current vPartition" on page 455
- "Get Detailed Information About vPartition" on page 455
- "Get Fiber Channel Adapters" on page 459
- "Get Disk Devices" on page 460
- "Get Network Interfaces" on page 461
- "Get File Systems" on page 462
- "Get Logical Volumes, Volume Groups, and Physical Volumes" on page 462
- "Get Network Interfaces" on page 465
- "Get Information About Link Aggregation Interfaces" on page 466
- "Get MAC Addresses of the Aggregated Interfaces" on page 466
- "Get Hardware Paths of the Aggregated Interfaces" on page 467
- "Get IP Addresses of the Aggregated Interfaces" on page 468

Verify Discovery on the vPartition

Goal	<ol style="list-style-type: none"> 1 To verify if discovery has connected to the vPartition. 2 To verify that further commands produce supported output.
Command	vparstatus -V
Output	Version 2.0
Values taken	<ol style="list-style-type: none"> 1 2.0. The version of the vparstatus executable 2 Return code
Comment	Supported versions of output are 2.0 and 1.3

Verify Discovery on the nPartition

Goal	To understand if discovery has connected to the partitionable server.
Command	parstatus -s
Output	None
Values taken	Return code
Comment	If return code is 0 , discovery has connected to the partitionable system

Get Information about Complex

Goal	To retrieve properties of the HP Complex CIT .
Command	parstatus -X
Output rp8420	<p>[Complex]</p> <p>Complex Name : Complex 01</p> <p>Complex Capacity</p> <p> Compute Cabinet (4 cell capable) : 1</p> <p>Active GSP Location : cabinet 0</p> <p>Model : 9000/800/rp8420</p> <p>Serial Number : DEH45419K0</p> <p>Current Product Number : A6912A</p> <p>Original Product Number : A6912A</p> <p>Complex Profile Revision : 1.0</p> <p>The total number of Partitions Present : 2</p>

Output rx8640	<p>[Complex]</p> <p>Complex Name : Complex 01</p> <p>Complex Capacity</p> <p> Compute Cabinet (4 cell capable) : 1</p> <p>Active MP Location : cabinet 0</p> <p>Original Product Name : server rx8640</p> <p>Original Serial Number : DEH4831H1Y</p> <p>Current Product Order Number : AB297A</p> <p>OEM Manufacturer :</p> <p>Complex Profile Revision : 1.0</p> <p>The total number of partitions present : 1</p>
Values taken	<ul style="list-style-type: none"> ▶ Complex Name > name ▶ Serial number/Original Serial Number > serialnumber, hostkey
Comment	HP Complex CIT derives from the Host CIT

List General Information About All Cells

Goal	To retrieve the list of names of all Cells of all Cabinets in the Complex.
Command	parstatus -C -M
Output rp8420	cell: cab0, cell0: active core :8/0/8 :48.0/ 0.0: cab0, bay0, chassis0 :yes :yes :0 cell: cab0, cell1: active core :4/0/8 :32.0/ 0.0: cab0, bay0, chassis1 :yes :yes :1 cell: cab0, cell2: active base :8/0/8 :40.0/ 0.0:- :no :yes :0 cell: cab0, cell3: active base :4/0/8 :32.0/ 0.0:- :no :yes :1
Output rx8640	cell: cab0, cell0: Active Core :8/0/8 :80.0/0.0 :cab0, bay0, chassis0 :yes :yes :0 cell: cab0, cell1: Active Base :8/0/8 :80.0/0.0 :cab0, bay0, chassis1 :yes :yes :0 cell: cab0, cell2: Active Base :4/0/8 :64.0/0.0 :- :no :yes :0 cell: cab0, cell3: Absent :- :- :- :-
Values taken	The names of the cells
Comment	The cell names are then used to retrieve detailed information about each cell.

List Detailed Information About Each Cell

Goal	To retrieve the properties of the Cell CIs and corresponding CPU and Memory CIs.
Command	parstatus -v -c <cell_number>

Output rp8420	<pre> [Cell] Hardware Location : cab0,cell0 Global Cell Number : 0 Actual Usage : active core Normal Usage : base Connected To : cab0,bay0,chassis0 Core Cell Capable : yes Firmware Revision : 24.1 Failure Usage : activate Use On Next Boot : yes Partition Number : 0 Partition Name : db01_ap02_db03_db04 [CPU Details] Type : 88E0 Speed : 1100 MHz CPU Status === ===== 0 ok 1 ok 2 ok 3 ok 4 ok 5 ok 6 ok 7 ok CPUs ===== OK : 8 Deconf : 0 Max : 8 </pre>
--------------------------	---

<p>Output rp8420 <i>(cont'd)</i></p>	<pre>[Memory Details] DIMM Size (MB) Status ===== 0A 4096 ok 4A 4096 ok 0B 4096 ok 4B 4096 ok 1A 4096 ok 5A 4096 ok 1B 4096 ok 5B 4096 ok 2A 4096 ok 2B 4096 ok 3A 4096 ok 3B 4096 ok Memory ===== DIMM OK : 12 DIMM Deconf : 0 Max DIMMs : 16 Memory OK : 48.00 GB Memory Deconf : 0.00 GB</pre>
--	---

Output rx8640	<pre> [Cell] Hardware Location : cab0,cell0 Global Cell Number : 0 Actual Usage : Active Core Normal Usage : Base Connected To : cab0,bay0,chassis0 Core Cell Capable : yes Firmware Revision : 9.48 Failure Usage : Normal Use On Next Boot : yes Partition Number : 0 Partition Name : db10_ap13_ap14_db15_db16_ap17_ap18_ap20 Requested CLM value : 0.0 GB Allocated CLM value : 0.0 GB Cell Architecture Type : Itanium(R)-based CPU Compatibility : CDH-640 Hyperthreading Capable : yes [CPU Details] Type : FFFF Speed : 1598 MHz CPU Status === ===== 0 OK 1 OK 2 OK 3 OK 4 OK 5 OK 6 OK 7 OK </pre>
----------------------	--

```
Output rx8640
(cont'd)

CPUs
=====
OK   : 8
Deconf : 0
Max   : 8

[Memory Details]

DIMM Size (MB) Status
=====
3A 8192 OK
3B 8192 OK
1A 8192 OK
1B 8192 OK
4A 8192 OK
4B 8192 OK
0A 8192 OK
0B 8192 OK
2A 8192 OK
2B 8192 OK

Memory
=====
DIMM OK   : 10
DIMM Deconf : 0
Max DIMMs : 16
Memory OK  : 80.00 GB
Memory Deconf : 0.00 GB
```


Values taken	Global Cell Number > name		
	Hardware Location > hardware_path		
	Actual Usage > is_core		If value of Actual Usage contains the word Core
	Core Cell Capable > core_capable		Convert yes/no to Boolean
	Requested CLM value > requested_clm_value		<ul style="list-style-type: none"> ▶ This parameter does not exist for rp8420 servers ▶ Need to convert GB to MB
	Allocated CLM value > allocated_clm_memory		<ul style="list-style-type: none"> ▶ This parameter does not exist for rp8420 servers ▶ Need to convert GB to MB
	Use On Next Boot > use_on_next_boot		Convert yes/no to Boolean
	Failure Usage > failure_usage		
	Firmware Revision > firmware_revision		
	Cell Architecture Type > architecture_type		This value does not exist for rp8420 servers
	CPU Compatibility > cpu_compatibility		This value does not exist for rp8420 servers
	Hyperthreading Capable > is_hyperthreading_capable		Convert yes/no to Boolean
	CPUs =====	deconf_cpu_number: 0	
	OK : 8	max_cpu_number: 8	
Deconf : 0			
Max : 8			

Values taken <i>(cont'd)</i>	Memory ===== DIMM OK : 10 DIMM Deconf : 0 Max DIMMs : 16 Memory OK : 80.00 GB Memory Deconf : 0.00 GB	memory_ amount: 80.00 GB deconf_ memory: 0.00 GB max_ dimms :16 deconfigure d_ dimms: 0	Need to convert GB to MB
Comment	The Memory CI is not created for UCMDB 9.x since there is no such CIT. The partition number is used to connect the cell to the nPartition (represented as a host).		

Get Information About I/O Chassis

Goal	To retrieve the data of all I/O chassis in the Complex (including I/O extension cabinets).	
Command	parstatus -I -M	
Output rp8420	chassis: cab0, bay0, chassis0 :active :yes :cab0, cell0:0 chassis: cab0, bay0, chassis1 :active :yes :cab0, cell1:1	
Output rx8640	chassis: cab0, bay0, chassis0 :Active :yes :cab0, cell0:0 chassis: cab0, bay0, chassis1 :Active :yes :cab0, cell1:0	
Values taken	name: cab0, bay0, chassis0	
	usage: Active	
	is_core: yes	To convert to Boolean values.
Comment	The Cell hardware path is used to connect the chassis to the Cell.	

Get the List of Names of the nPartitions on the System

Goal	To retrieve the list of the nPartition numbers configured on the system.
Command	parstatus -P -M
Output rp8420	partition: 0 :active : 2 : 1 :cab0,cell0:db01_ap02_db03_db04 partition: 1 :active : 2 : 1 :cab0,cell1:wdb1_wdb4
Output rx8640	partition:0 :Active :3 :2 :cab0,cell0:db10_ap13_ap14_db15_db16_ap17_
Values taken	The list of nPartition numbers
Comment	These numbers are used to retrieve detailed information about each nPartition.

Get Detailed Information About nPartition

Goal	To retrieve detailed information for each nPartition and create a Host, connected to the Cells and to the HP nPar Config CI .
Command	parstatus -v -p <npartition_number>

```

Output
rp8420
[Partition]
Partition Number      : 0
Partition Name       : db01_ap02_db03_db04
Status                : active
IP address            : 0.0.0.0
Primary Boot Path    : 0/0/0/2/0.6.0
Alternate Boot Path  : 0/0/0/2/1.2.0
HA Alternate Boot Path : 0/0/0/3/0.6.0
PDC Revision         : 24.1
IODCH Version        : 88E0
CPU Speed            : 1100 MHz
Core Cell            : cab0,cell0

[Cell]
                CPU   Memory           Use
                OK/  (GB)           Core On
Hardware Actual Deconf/ OK/           Cell Next Par
Location Usage  Max  Deconf  Connected To  Capable
Boot Num
=====
cab0,cell0 active core 8/0/8  48.0/ 0.0 cab0,bay0,chassis0 yes
yes 0
cab0,cell2 active base 8/0/8  40.0/ 0.0 -          no  yes 0

[Chassis]
                Core Connected Par
Hardware Location Usage  IO To  Num
=====
cab0,bay0,chassis0 active  yes cab0,cell0 0
    
```

Output rx8640	<pre> [Partition] Partition Number : 0 Partition Name : db10_ap13_ap14_db15_db16_ap17_ap18_ap20 Status : Active IP Address : Primary Boot Path : 0/0/8/1/0/4/0.8.0.255.0.12.0 Alternate Boot Path : 0/0/8/1/0/4/1.8.0.255.0.13.0 HA Alternate Boot Path : PDC Revision : 9.48 IODCH Version : fff Cell Architecture : Itanium(R)-based CPU Compatibility : CDH-640 CPU Speed : 1598 MHz Core Cell : cab0,cell0 Core Cell Choice [0] : cab0,cell0 Total Good Memory Size : 224.0 GB Total Interleave Memory: 224.0 GB Total Requested CLM : 0.0 GB Total Allocated CLM : 0.0 GB Hyperthreading Enabled : no [Cell] </pre> <table border="1"> <thead> <tr> <th></th> <th>CPU</th> <th>Memory</th> <th></th> <th>Use</th> <th></th> </tr> <tr> <th></th> <th>OK/</th> <th>(GB)</th> <th></th> <th>Core</th> <th>On</th> </tr> <tr> <th>Hardware Location</th> <th>Actual Usage</th> <th>Deconf/ Max</th> <th>OK/ Deconf</th> <th>Cell Connected To</th> <th>Next Par Capable</th> </tr> <tr> <th></th> <th>Boot Num</th> <th></th> <th></th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td colspan="6">=====</td> </tr> <tr> <td>cab0,cell0</td> <td>Active</td> <td>Core</td> <td>8/0/8</td> <td>80.0/0.0</td> <td>cab0,bay0,chassis0</td> <td>yes</td> </tr> <tr> <td></td> <td>yes</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>cab0,cell1</td> <td>Active</td> <td>Base</td> <td>8/0/8</td> <td>80.0/0.0</td> <td>cab0,bay0,chassis1</td> <td>yes</td> </tr> <tr> <td></td> <td>yes</td> <td>0</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		CPU	Memory		Use			OK/	(GB)		Core	On	Hardware Location	Actual Usage	Deconf/ Max	OK/ Deconf	Cell Connected To	Next Par Capable		Boot Num					=====						cab0,cell0	Active	Core	8/0/8	80.0/0.0	cab0,bay0,chassis0	yes		yes	0					cab0,cell1	Active	Base	8/0/8	80.0/0.0	cab0,bay0,chassis1	yes		yes	0				
	CPU	Memory		Use																																																							
	OK/	(GB)		Core	On																																																						
Hardware Location	Actual Usage	Deconf/ Max	OK/ Deconf	Cell Connected To	Next Par Capable																																																						
	Boot Num																																																										
=====																																																											
cab0,cell0	Active	Core	8/0/8	80.0/0.0	cab0,bay0,chassis0	yes																																																					
	yes	0																																																									
cab0,cell1	Active	Base	8/0/8	80.0/0.0	cab0,bay0,chassis1	yes																																																					
	yes	0																																																									

<p>Output rx8640 (cont'd)</p>	<pre> cab0,cell2 Active Base 4/0/8 64.0/0.0 - no yes 0 Notes: * = Cell has no interleaved memory. [Chassis] Core Connected Par Hardware Location Usage IO To Num ===== cab0,bay0,chassis0 Active yes cab0,cell0 0 [Chassis] Core Connected Par Hardware Location Usage IO To Num ===== cab0,bay0,chassis1 Active yes cab0,cell1 0 </pre>																	
<p>Values taken</p>	<table border="1"> <tr> <td colspan="2" data-bbox="491 887 1229 939">Host (nPartition)</td> </tr> <tr> <td data-bbox="491 939 833 1055">hostkey</td> <td data-bbox="833 939 1229 1055">Host key is composed of nPartition name and Complex Serial number</td> </tr> <tr> <td data-bbox="491 1055 833 1107">Partition Name > tname</td> <td data-bbox="833 1055 1229 1107"></td> </tr> <tr> <td colspan="2" data-bbox="491 1107 1229 1159">HP nPar Config</td> </tr> <tr> <td data-bbox="491 1159 833 1237">Constant "nPar Config" > name</td> <td data-bbox="833 1159 1229 1237"></td> </tr> <tr> <td data-bbox="491 1237 833 1321">Partition Name > npar_name</td> <td data-bbox="833 1237 1229 1321"></td> </tr> <tr> <td data-bbox="491 1321 833 1373">Status > npar_status</td> <td data-bbox="833 1321 1229 1373"></td> </tr> <tr> <td data-bbox="491 1373 833 1453">PDC Revision > pdc_revision</td> <td data-bbox="833 1373 1229 1453"></td> </tr> </table>		Host (nPartition)		hostkey	Host key is composed of nPartition name and Complex Serial number	Partition Name > tname		HP nPar Config		Constant "nPar Config" > name		Partition Name > npar_name		Status > npar_status		PDC Revision > pdc_revision	
Host (nPartition)																		
hostkey	Host key is composed of nPartition name and Complex Serial number																	
Partition Name > tname																		
HP nPar Config																		
Constant "nPar Config" > name																		
Partition Name > npar_name																		
Status > npar_status																		
PDC Revision > pdc_revision																		

Values <i>(cont'd)</i>	Hyperthreading Enabled > hyperthreading_mode	This value does not exist on the rp8420 servers
	Partition Number > partition_number	
	Primary Boot Path > primary_boot_path	
	Alternate Boot Path > alternate_boot_path	

Get the Name of the Current vPartition

Goal	To retrieve the name of the current vPartition.
Command	vparstatus -w -M
Output	doidb01
Values taken	The name of the vPartition that discovery has connected to.
Comment	The list includes detailed information for the current vPartition only. It is possible to retrieve detailed information about all vPartitions on the nPartition, but it is not possible to retrieve their IP addresses and/or lower MAC address to create a host in UCMDB.

Get Detailed Information About vPartition

Goal	To retrieve detailed information about vPartition and create Host and HP vPar Config CIs.
Command	vparstatus -v -p <vpartition_name>

<p>Output rp8420</p>	<pre> [Virtual Partition Details] Name: doidb01 State: Up Attributes: Dynamic,Autoboot,Nosearch Kernel Path: /stand/vmunix Boot Opts: -lq [CPU Details] Min/Max: 3/16 Bound by User [Path]: 0.15 0.16 0.17 Bound by Monitor [Path]: Unbound [Path]: 2.14 2.15 [IO Details] 0.0.12 0.0.14 0.0.12.1.0.4.0.8.0.255.0.0.0 0.0.14.1.0.4.0.8.0.255.0.1.0 0.0.12.1.0.4.0.111.128.19.4.0.0 0.0.12.1.0.4.0.111.88.19.5.0.0 BOOT 0.0.14.1.0.4.0.112.88.19.5.0.0, ALTBOOT [Memory Details] Specified [Base /Range]: (bytes) (MB) Total Memory (MB): 24448 </pre>
-----------------------------	---

Output rx8640	<pre> [Virtual Partition Details] Name: doiap17 State: Up Attributes: Dynamic,Autoboot,Nosearch Kernel Path: /stand/vmunix Boot Opts: -lq [CPU Details] Min/Max: 1/12 User assigned [Path]: Boot processor [Path]: 1.122 Monitor assigned [Path]: Non-cell-specific: User assigned [Count]: 1 Monitor assigned [Count]: 0 Cell-specific [Count]: Cell ID/Count <none> [IO Details] 0.0.8 0.0.8.1.0.4.0.8.0.255.0.13.0 0.0.8.1.0.4.0.8.0.255.0.12.0 BOOT 0.0.8.1.0.4.1.8.0.255.0.13.0,ALTBOOT [Memory Details] ILM, user-assigned [Base /Range]: (bytes) (MB) ILM, monitor-assigned [Base /Range]: 0x11c0000000/8192 (bytes) (MB) ILM Total (MB): 8192 </pre>
----------------------	---

<p>Output rx8640 (cont'd)</p>	<p>ILM Granularity (MB): 512</p> <p>CLM, user-assigned [CellID Base /Range]: (bytes) (MB)</p> <p>CLM, monitor-assigned [CellID Base /Range]: (bytes) (MB)</p> <p>CLM (CellID MB):</p> <p>CLM Granularity (MB): 128</p>	
<p>Values taken</p>	<p>Const "HP vPar Config" > name</p>	
	<p>Name > vpar_name</p>	
	<p>Boot Opts > boot_options</p>	
	<p>Boot processor [Path] > boot_processor_path</p>	<p>This value does not exist for rp8420 servers</p>
	<p>State > vpar_status</p>	
	<p>Attributes: Dynamic, Autoboot, Nosearch</p>	<ul style="list-style-type: none"> ▶ autoboot_mode: Autoboot ▶ autosearch_mode: Nosearch ▶ modification_mode: Dynamic
	<p>Bound by User [Path]/User assigned [Path] > cpus_bound_by_user</p>	<p>Actual parameter is different between server versions</p>
	<p>Unbound [Path] > unbound_cpus</p>	
<p>Comment</p>	<p>For the attribute format of attributes such as cpus_bound_by_user, refer to the Data Model specification.</p>	

Get Fiber Channel Adapters

Goal	To model Fibre Channel adapters	
Command	ioscan -FnkCfc	
Output	<pre>pci:wsio:F:T:F:-1:50:4294967295:fc:fcd:0/0/12/1/0/4/0:16 119 35 18 0 0 0 0 :0:root.cell.sba.lba.PCItoPCI.fcd:fcd:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 1):0 /dev/fcd0 pci:wsio:F:T:F:-1:50:4294967295:fc:fcd:0/0/12/1/0/4/1:16 119 35 18 0 0 0 0 :1:root.cell.sba.lba.PCItoPCI.fcd:fcd:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2):1 /dev/fcd1 pci:wsio:F:T:F:-1:50:4294967295:fc:fcd:0/0/14/1/0/4/0:16 119 35 18 0 0 0 0 :2:root.cell.sba.lba.PCItoPCI.fcd:fcd:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 1):2 /dev/fcd2 pci:wsio:F:T:F:-1:50:4294967295:fc:fcd:0/0/14/1/0/4/1:16 119 35 18 0 0 0 0 :3:root.cell.sba.lba.PCItoPCI.fcd:fcd:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2):3 /dev/fcd3</pre>	
Values taken	name	/dev/fcd0
	data_description	HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2)
Comment	The hardware path serves to locate the Cell and use it as a container for FC HBA. Example value: 0/0/14/1/0/4/0. The first integer value is the Global ID of the Cell; the second value is the ID of the I/O chassis.	

Get Disk Devices

Goal	To retrieve information about the dependency between I/O chassis, physical disk, and SCSI adapter.	
Command	ioscan -FnkCdisk	
Output	<pre>scsi:wsio:T:T:F:31:188:2031616:disk:sdisk:0/0/12/1/0/4/0.111.88.19.5 .0.0:0 0 4 50 0 0 0 0 51 248 164 14 99 72 178 210 :3:root.cell.sba.lba.PCItoPCI.fcd.fcd_fcp.fcd_vbus.tgt.sdisk:sdisk:CL AIMED:DEVICE:EMC SYMMETRIX:31 /dev/dsk/c31t0d0 /dev/rdisk/c31t0d0 scsi:wsio:T:T:F:31:188:2031872:disk:sdisk:0/0/12/1/0/4/0.111.88.19.5 .0.1:0 0 4 50 0 0 0 0 51 248 164 14 76 238 217 30 :59:root.cell.sba.lba.PCItoPCI.fcd.fcd_fcp.fcd_vbus.tgt.sdisk:sdisk:CL AIMED:DEVICE:EMC SYMMETRIX:31 /dev/dsk/c31t0d1 /dev/rdisk/c31t0d1 scsi:wsio:T:T:F:31:188:2032128:disk:sdisk:0/0/12/1/0/4/0.111.88.19.5 .0.2:0 0 4 50 0 0 0 0 51 248 164 14 101 17 172 238 :61:root.cell.sba.lba.PCItoPCI.fcd.fcd_fcp.fcd_vbus.tgt.sdisk:sdisk:CL AIMED:DEVICE:EMC SYMMETRIX:31 /dev/dsk/c31t0d2 /dev/rdisk/c31t0d2</pre>	
Values taken	slot_number	0/0/12/1/0/4/0.111.88.19.5.0.0
	name	/dev/dsk/c31t0d2
	Cell ID	0/0/12/1/0/4/0.111.88.19.5.0.0
	IO chassis ID	0/0/12/1/0/4/0.111.88.19.5.0.0

Get Network Interfaces

Goal	To retrieve information about the dependency between network interfaces and the I/O chassis.
Command	<code>ioscan -FnkClan</code>
Output	<pre>pci:wsio:F:F:F:-1:- 1:4294967295:lan:igelan:0/0/12/1/0/6/0:20 228 22 72 0 0 0 0 :0:root.cell.sba.lba.PCItoPCI.igelan:igelan:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:0 pci:wsio:F:F:F:-1:- 1:4294967295:lan:igelan:0/0/12/1/0/6/1:20 228 22 72 0 0 0 0 :1:root.cell.sba.lba.PCItoPCI.igelan:igelan:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:1 pci:wsio:F:F:F:-1:- 1:4294967295:lan:igelan:0/0/14/1/0/6/0:20 228 22 72 0 0 0 0 :2:root.cell.sba.lba.PCItoPCI.igelan:igelan:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:2 pci:wsio:F:F:F:-1:- 1:4294967295:lan:igelan:0/0/14/1/0/6/1:20 228 22 72 0 0 0 0 :3:root.cell.sba.lba.PCItoPCI.igelan:igelan:CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:3</pre>
Values taken	The hardware path which reflects the Cell and I/O chassis that this interface belongs to.

Get File Systems

Goal	To retrieve information about the file systems and corresponding logical volumes.
Command	df -P
Output	<pre>Filesystem 512-blocks Used Available Capacity Mounted on /dev/vg01/lv106 9837710 115094 9722616 2% /usr/vw/rvs /dev/vg01/lv124 7915344 814616 7100728 11% /home/kdov12 /dev/vg01/lv125 10222640 6275190 3947450 62% /home/ebrev /dev/vg01/lv123 20829536 2796208 18033328 14% /home/temp /dev/vg01/lv110 2080832 4608 2076224 1% /oracle2/arch/inst_aebp</pre>
Values taken	name for FileSystem CIT: /usr/vw/rvs Name of the logical volume: /dev/vg01/lv106

Get Logical Volumes, Volume Groups, and Physical Volumes

Goal	To retrieve data for modeling Logical volumes, Volume groups, and Physical volumes.
Command	vgdisplay -v

Output	--- Volume groups ---
	VG Name /dev/vg00
	VG Write Access read/write
	VG Status available
	Max LV 255
	Cur LV 10
	Open LV 10
	Max PV 16
	Cur PV 1
	Act PV 1
	Max PE per PV 4384
	VGDA 2
	PE Size (Mbytes) 16
	Total PE 4315
	Alloc PE 4156
	Free PE 159
	Total PVG 0
	Total Spare PVs 0
	Total Spare PVs in use 0
	--- Logical volumes ---
	LV Name /dev/vg00/lvol1
	LV Status available/syncd
	LV Size (Mbytes) 256
	Current LE 16
	Allocated PE 16
	Used PV 1

<p>Output <i>(cont'd)</i></p>	<pre> --- Physical volumes --- PV Name /dev/dsk/c31t0d0 PV Name /dev/dsk/c32t0d0 Alternate Link PV Status available Total PE 4315 Free PE 159 Autoswitch On Proactive Polling On </pre>																			
<p>Values taken</p>	<table border="1"> <tr> <td colspan="2" data-bbox="531 499 1232 552">Volume group</td> </tr> <tr> <td data-bbox="531 552 852 678"> VG Name > name VG Write Access > write_access </td> <td data-bbox="852 552 1232 678"></td> </tr> <tr> <td data-bbox="531 678 852 774"> VG Status > vg_status PE Size (Mbytes) </td> <td data-bbox="852 678 1232 774"> This value is used to calculate the size of the physical volume </td> </tr> <tr> <td colspan="2" data-bbox="531 774 1232 826">Logical Volume</td> </tr> <tr> <td data-bbox="531 826 852 916"> LV Name > name LV Status > lv_status </td> <td data-bbox="852 826 1232 916"></td> </tr> <tr> <td colspan="2" data-bbox="531 916 1232 968">Physical Volume</td> </tr> <tr> <td data-bbox="531 968 852 1083"> PV Name > name </td> <td data-bbox="852 968 1232 1083"> Alternate link may also be used. It depends on the output of the ioscan FnkCdisk command. </td> </tr> <tr> <td data-bbox="531 1083 852 1135"> PV Status > pv_status </td> <td data-bbox="852 1083 1232 1135"></td> </tr> <tr> <td data-bbox="531 1135 852 1215"> Total PE > pv_size </td> <td data-bbox="852 1135 1232 1215"> This attribute is calculated on the PE Size (Mbytes) value. </td> </tr> </table>		Volume group		VG Name > name VG Write Access > write_access		VG Status > vg_status PE Size (Mbytes)	This value is used to calculate the size of the physical volume	Logical Volume		LV Name > name LV Status > lv_status		Physical Volume		PV Name > name	Alternate link may also be used. It depends on the output of the ioscan FnkCdisk command.	PV Status > pv_status		Total PE > pv_size	This attribute is calculated on the PE Size (Mbytes) value.
Volume group																				
VG Name > name VG Write Access > write_access																				
VG Status > vg_status PE Size (Mbytes)	This value is used to calculate the size of the physical volume																			
Logical Volume																				
LV Name > name LV Status > lv_status																				
Physical Volume																				
PV Name > name	Alternate link may also be used. It depends on the output of the ioscan FnkCdisk command.																			
PV Status > pv_status																				
Total PE > pv_size	This attribute is calculated on the PE Size (Mbytes) value.																			

Get Network Interfaces

Goal	To retrieve information about the network interfaces.
Command	lanscan
Output	<pre> Hardware Station Crd Hdw Net-Interface NM MAC HP-DLPI DLPI Path Address In# State NamePPA ID Type Support Mjr# 0/0/4/1/0/6/1 0x0014C254D9BD 1 UP lan1 snap1 2 ETHER Yes 119 0/0/6/1/0/6/1 0x0014C254C961 3 UP lan3 snap3 4 ETHER Yes 119 LinkAgg0 0x0014C254D9BC 900 UP lan900 snap900 6 ETHER Yes 119 LinkAgg1 0x000000000000 901 DOWN lan901 snap901 7 ETHER Yes 119 LinkAgg2 0x000000000000 902 DOWN lan902 snap902 8 ETHER Yes 119 LinkAgg3 0x000000000000 903 DOWN lan903 snap903 9 ETHER Yes 119 LinkAgg4 0x000000000000 904 DOWN lan904 snap904 10 ETHER Yes 119 </pre>
Values taken	<ul style="list-style-type: none"> ▶ The hardware path to create the link between the network interface and I/O chassis. ▶ The MAC address to create the network interface. ▶ The MAC address of the Link aggregation interface, the indicator that the interface is up, and the device name.

Get Information About Link Aggregation Interfaces

Goal	To model the links between interfaces and link aggregation.
Command	lanscan -q
Output	1 3 900 0 2 901 902 903 904
Values taken	The interface number and IDs of the aggregated interfaces.

Get MAC Addresses of the Aggregated Interfaces

Goal	To retrieve the MAC addresses of the aggregated interfaces.
Command	lanadmin -a <interface_id>
Example	lanscan -a 0
Output	Station Address = 0x0014c254d9bc
Values taken	The MAC address of the aggregated interface

Get Hardware Paths of the Aggregated Interfaces

Goal	To retrieve the hardware path of the aggregated interfaces
Command	<code>lanscan -v grep -E <list_of_aggregated_interfaces></code>
Example	<code>lanscan -v grep -E "lan0 lan2"</code>
Output	<pre>0/0/4/1/0/6/0 0 UP lan0 snap0 1 ETHER Yes 119 igelan 0/0/6/1/0/6/0 2 UP lan2 snap2 3 ETHER Yes 119 igelan</pre>
Values taken	The hardware path that allocates the I/O chassis that holds this interface.

Get IP Addresses of the Aggregated Interfaces

Goal	To get IP addresses of the interfaces
Command	netstat -rn
Output	<pre> Routing tables Destination Gateway Flags Refs Interface Pmtu 127.0.0.1 127.0.0.1 UH 0 lo0 4136 10.186.112.115 10.186.112.115 UH 0 lan0 4136 10.186.116.13 10.186.116.13 UH 0 lan1 4136 192.168.121.1 192.168.121.1 UH 0 lan2 4136 10.186.115.18 10.186.115.18 UH 0 lan3 4136 10.186.116.19 10.186.116.19 UH 0 lan1:1 4136 10.186.116.0 10.186.116.13 U 3 lan1 1500 10.186.116.0 10.186.116.19 U 3 lan1:1 1500 10.186.115.0 10.186.115.18 U 2 lan3 1500 10.186.112.0 10.186.112.115 U 2 lan0 1500 192.168.121.0 192.168.121.1 U 2 lan2 1500 10.186.86.0 10.186.115.1 UG 0 lan3 1500 127.0.0.0 127.0.0.1 U 0 lo0 4136 default 10.186.116.1 UG 0 lan1 1500 </pre>
Values taken	<p>The IP addresses of the interfaces.</p> <p>The netstat command does not require root privileges, in contrast to ifconfig.</p>

Troubleshooting and Limitations

- The destination host is not a part of the HP nPartition system.
DFM considers the target host as not being a part of the HP partitionable system. The criteria are based on executing the **parstatus -s** command.
- Failed to discover vPartition details.
The **vparstatus** command was not executed successfully. This command should be accessible and DFM should have enough permissions to execute it. If this command requires **sudo** to be executed, configure the SSH credentials. For credentials information, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- Failed to discover storage topology.
The **vgdisplay** command was not executed successfully.
- Failed to link file systems and disks.
The **df** command was not executed successfully.
- Failed to discover SCSI adapters.
Failed to discover Fibre Channel adapters.
Failed to discover Network cards.
The **ioscan** command was not executed successfully.

34

IBM HMC

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 472

Tasks

- ▶ Discover IBM HMC on page 473

Reference

- ▶ The IBM HMC by Shell Job on page 480
- ▶ The IBM LPar and VIO by Shell Job on page 480
- ▶ The IBM_HMC_SHELL_PATTERN Adapter on page 481
- ▶ The IBM_LPAR_VIO_BY_SHELL Adapter on page 482
- ▶ Discovery Mechanism on page 483
- ▶ VIO Server Side Commands on page 498
- ▶ LPAR Side Commands on page 510

Troubleshooting and Limitations on page 511

Concepts

Overview

This document describes the usage and functionality of the IBM HMC discovery package.

Hardware Management Console is a technology invented by IBM for the purpose of providing a standard interface for configuring and operating partitioned (also known as an LPAR or virtualized system) and SMP systems such as IBM System I or IBM System p series.

Tasks

Discover IBM HMC

This task includes the following steps:

- "Supported Versions" on page 473
- "Prerequisites" on page 473
- "Set up Protocols" on page 475
- "Deploy the Package" on page 475
- "Discovery Workflow" on page 475
- "Created/Changed Entities" on page 476
- "Sample Output" on page 478
- "Discovered CITs" on page 479

1 Supported Versions

This discovery solution supports the IBM Power 4 and Power 5 Series on AIX and Linux.

2 Prerequisites

This discovery solution is based on the SSH and Telnet Shell protocols.

Before activating discovery, confirm that the discovery user has all the required permissions to run the following commands:

Command	For Details, See:
lscfg	page 510
lsdev -dev <Device>	
lshmc -b	page 485
lshmc -n	page 486

Command	For Details, See:
lshmc -v	page 485
lshmc -V	page 484
lshwres -r io --subtype slot -m <pSeriesName>	page 497
lshwres -r mem --level lpar -m <lparName>	
lshwres -r mem --level sys -m <pSeriesName>	page 489
lshwres -r proc --level lpar -m <lparName>	page 496
lshwres -r proc --level pool -m <pSeriesName>	page 490
lshwres -r proc --level sys -m <pSeriesName>	page 488
lshwres -r virtualio --subtype eth --level lpar -m <LParName>	page 494
lshwres -r virtualio --subtype scsi -m <LPar Name>	page 495
lslv	
lslv -v <Logical Volume Name>	page 507
lsmmap -all	page 508
lsmmap -all -net	page 502
lspartition	
lspath	
lspv	
lssyscfg -r lpar -m <LPar Name>	
lssyscfg -r prof -m <LPar Name>	
lssyscfg -r sys	page 487
lsvg	page 503
lsvg -l <Volume Group Name>	page 505
lsvio -e	
lsvio -s	
lvdisplay	

Command	For Details, See:
pvdisplay	
vgdisplay	

3 Set up Protocols

For credentials information, see:

- "SSH Protocol"
- "Telnet Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

If some of the commands are configured to run with **sudo** on the target host, in the **Protocol Parameters** dialog box, fill in the following fields:

- **Sudo paths.** Enter the full path to the sudo executable, together with the name of the executable. You can add more than one entry if executable files are placed in various places on the target operating systems.

Example: sudo,/usr/bin/sudo,/bin/sudo

- **Sudo commands.** Enter a list of commands that are prefixed with **sudo**.

Example: lspath,ifconfig

For details, see "Protocol Parameter Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

4 Deploy the Package

For details on deploying packages, see "Package Manager" in the *HP Universal CMDB Administration Guide*.

5 Discovery Workflow

- a Run the **Range IPs by ICMP** job.
- b Run the **Host Connection by Shell** job.
- c Run the **IBM HMC by Shell** job.
- d Run the **IBM LPar and VIO by Shell** job.

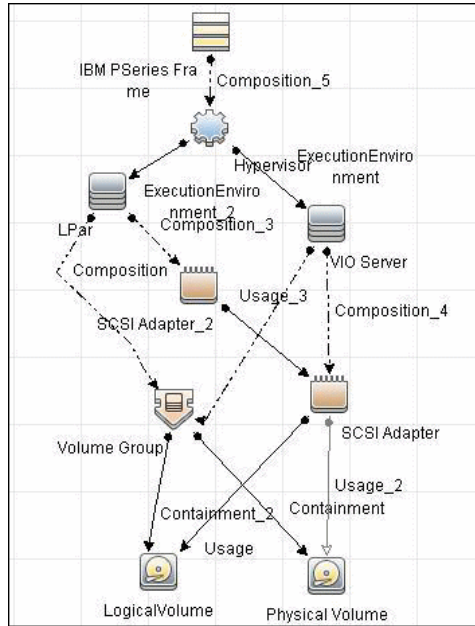
For details on running jobs, refer to the "Discovery Control Panel" chapter in *HP Universal CMDB Data Flow Management Guide*.

6 Created/Changed Entities

Entity Name	Entity Type	Entity Description
IBM HMC	CI Type	HMC software
IBM LPar Profile	CI Type	LPar configuration
IBM Processor Pool	CI Type	Shared Processor Pool
IBM PSeries Frame	CI Type	PSeries Frame/Managed System
Interface Aggregation	CI Type	Link Aggregation
I/O Slot	CI Type	I/O Slot on the Frame
SEA Adapter	CI Type	Virtual Eth interface on a VIO Server
IBM Processor Pool > containment > CPU	Valid Link	
I/O Slot > containment > Fiber Channel HBA	Valid Link	
I/O Slot > containment > Network Interface	Valid Link	
I/O Slot > containment > SCSI Adapter	Valid Link	
IBM HMC > manage > IBM PSeries Frame	Valid Link	
Interface Aggregation > membership > Network Interface	Valid Link	
Network Interface > realization > Network Interface	Valid Link	
Network Interface > usage > SEA Adapter	Valid Link	
SEA Adapter > usage > Network Interface	Valid Link	
IBM HMC by Shell	Job	Performs HMC based discovery
IBM LPAR and VIO Server Topology by Shell	Job	Performs LPAR and VIO Server side discovery
Virtualization - IBM HMC	Discovery Module	

Entity Name	Entity Type	Entity Description
IBM_HMC_BY_SHELL_PATTERN	Adapter	Adapter for the IBM HMC by Shell job
IBM_LPAR_VIO_BY_SHELL	Adapter	Adapter for the IBM LPAR and VIO Server Topology by Shell job
ibm_hmc_by_shell	Script	General HMC side discovery script
ibm_hmc_lib	Script	Common Data Objects and Procedures for both new Jobs
ibm_lpar_or_vio_by_shell	Script	General VIO Server and LPAR discovery script
ibm_hmc_by_shell.xml	query	Trigger query for the IBM HMC by Shell job
ibm_lpar_or_vio_trigger_tql.xml	query	Trigger query for the IBM LPAR and VIO Server Topology by Shell job
IBM HMC Topology.xml	query	Query (TQL) for the IBM HMC Topology view
IBM Storage Topology.xml	query	Query (TQL) for the IBM Storage Topology view
IBM HMC Topology.xml	View	
IBM Storage Topology.xml	View	
lpar_boot_mode	Type	Supported boot modes
lpar_cpu_mode	Type	CPU Sharing modes
lpar_sharing_mode	Type	LPAR cap/uncap sharing modes
lpar_state	Type	Possible LPAR states
lpar_type	Type	Possible LPAR types

IBM Storage Topology:

**8 Discovered CITs**

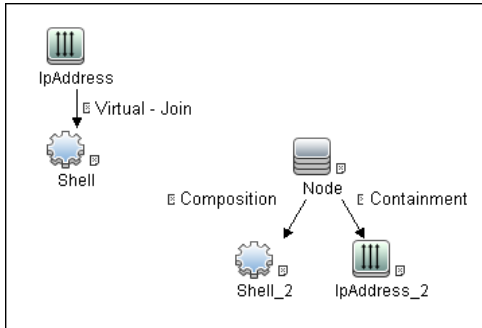
For details, see the following sections:

Adapter	For Details on Discovered CITs, See:
The IBM_HMC_SHELL_PATTERN Adapter	page 481
The IBM_LPAR_VIO_BY_SHELL Adapter	page 483

Reference

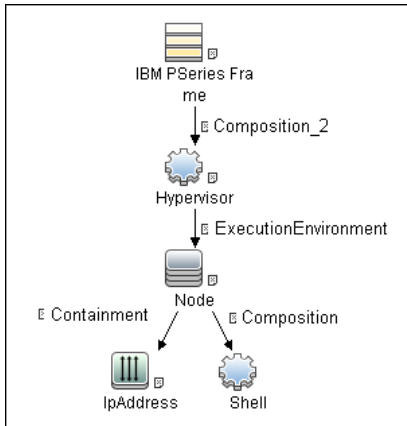
The IBM HMC by Shell Job

Trigger Query



The IBM LPar and VIO by Shell Job

Trigger Query



The IBM_HMC_SHELL_PATTERN Adapter

Input Query



Triggered CI Data

Triggered CI Data	
Name	
ip_address	\${SOURCE.ip_address}
ip_domain	\${SOURCE.ip_domain}

Discovered CITs

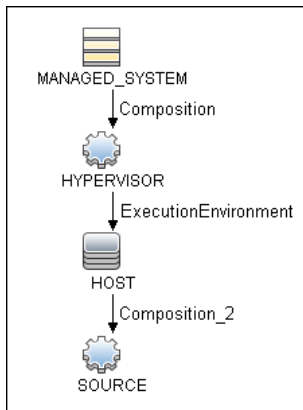
Composition
Containment
Cpu
ExecutionEnvironment
I/O Slot
IBM Frame
IBM HMC
IBM LPar Profile
IBM Processor Pool
Interface
IpAddress
Manage
Membership
Node
PhysicalPort
Realization
SCSI Adapter
Shell
Usage
Virtualization Layer Software
Man

Used Scripts

- ibm_hmc_by_shell.py
- storage_topology.py
- ibm_hmc_lib.py

The IBM_LPAR_VIO_BY_SHELL Adapter

Input Query



Triggered CI Data

Name	
Protocol	\${SOURCE.root_class}
credentialsId	\${SOURCE.credentials_id}
hostId	\${SOURCE.root_container}
ip_address	\${SOURCE.application_ip}
managedSystemId	\${MANAGED_SYSTEM.root_id}
osType	\${HOST.host_os}

Discovered CITs

Composition
Containment
Dependency
Fibre Channel HBA
FileSystem
I/O Slot
Interface
Interface Aggregation
Interface Index
IpAddress
LogicalVolume
Membership
Node
Parent
Physical Volume
Realization
SCSI Adapter
SEA Adapter
Usage
Volume Group

Used Scripts

- `ibm_lpar_or_vio_by_shell.py`
- `storage_topology.py`
- `ibm_hmc_lib.py`

Discovery Mechanism

This section includes the following commands:

- `"lshmc -V"` on page 484
- `"lshmc -v"` on page 485
- `"lshmc -b"` on page 485
- `"lshmc -n"` on page 486
- `"lspartition -c <TYPE>_<VERSION> -i"` on page 486

- "lssyscfg -r sys" on page 487
- "lshwres -r proc --level sys -m '<Managed System Name>'" on page 488
- "lshwres -r proc --level pool -m '<Managed System Name>'" on page 490
- "lssyscfg -r lpar -m '<Managed System Name>'" on page 491
- "lssyscfg -r prof -m '<Managed System Name>'" on page 492
- "lshwres -r virtualio --rsubtype eth --level lpar -m '<Managed System Name>'" on page 494
- "lshwres -r virtualio --rsubtype scsi -m '<Managed System Name>'" on page 495
- "lshwres -r proc --level lpar -m '<Managed System Name>'" on page 496
- "lshwres -r io --rsubtype slot -m '<Managed System Name>'" on page 497

lshmc -V

Output

```
version= Version: 7 Release: 3.5.0 Service Pack: 0 HMC Build level 20091201.1
MH01195: Required fix for HMC V7R3.5.0 (10-16-2009) MH01197: Fix for HMC
V7R3.5.0 (11-12-2009) MH01204: Fix for HMC V7R3.5.0 (12-11-2009)
", "base_version=V7R3.5.0 "
```

Mapping

The output of this command is used to fill in the attributes of the **IBM HMC** CI:

CMD Output Attribute	CI Name	CI Attribute
Version	IBM HMC	Version_number
Base_version	IBM HMC	Application_version_description

Ishmc -v

Output

```
vpd=*FC ????????? *VC 20.0 *N2 Tue Apr 27 13:05:33 GEST 2010 *FC ????????? *DS
Hardware Management Console *TM eserver xSeries 335 -[XXXXCR2]- *SE XXXXXXXX
*MN IBM *PN Unknown *SZ 1059495936 *OS Embedded Operating Systems *NA
192.168.1.10 *FC ????????? *DS Platform Firmware *RM V7R3.5.0.0
```

Mapping

The output of this command is used to fill in the attributes of the **IBM HMC** CI:

CMD Output Attribute	CI Name	CI Attribute
SE	IBM HMC	HMC Serial Number
TM	IBM HMC	HMC TYPE

Ishmc -b

Output

```
bios=T2E139AUS-1.15
```

Mapping

The output of this command is used to fill in the attributes of the **IBM HMC** CI:

CMD Output Attribute	CI Name	CI Attribute
Bios	IBM HMC	HMC BIOS

lshmc -n**Output**

```
hostname=hmc01,domain=somedomain.com,"ipaddr=192.168.1.10,0.0.0.0,192.168.128.1",
"networkmask=255.255.254.0,255.255.255.0,255.255.128.0",gateway=192.168.1.1,
nameserver=,domainsuffix=,slipipaddr=192.168.1.1,slipnetmask=255.255.0.0,"ipaddr
lpar=192.168.80.1,192.168.128.1","networkmasklpar=255.255.254.0,255.255.128.0",cli
ents=,ipv6addrlpar=,ipv4addr_eth0=192.168.1.10,ipv4netmask_eth0=255.255.254.0,ip
v4dhcp_eth0=off,ipv6addr_eth0=,ipv6auto_eth0=off,ipv6privacy_eth0=off,ipv6dhcp_eth
0=off,lparcomm_eth0=off,jumboframe_eth0=off,speed_eth0=100,duplex_eth0=full,tso_
eth0=off,ipv4addr_eth1=0.0.0.0,ipv4netmask_eth1=255.255.255.0,ipv4dhcp_eth1=off,i
pv6addr_eth1=,ipv6auto_eth1=off,ipv6privacy_eth1=off,ipv6dhcp_eth1=off,lparcomm_
eth1=off,jumboframe_eth1=off,speed_eth1=auto,duplex_eth1=auto,tso_eth1=off,ipv4a
ddr_eth2=192.168.128.1,ipv4netmask_eth2=255.255.128.0,ipv4dhcp_eth2=off,ipv6add
r_eth2=,ipv6auto_eth2=off,ipv6privacy_eth2=off,ipv6dhcp_eth2=off,lparcomm_eth2=off
,jumboframe_eth2=off,speed_eth2=auto,duplex_eth2=auto,tso_eth2=off
```

Mapping

The output of this command is used to fill in the network information for a particular HMC machine. A host with HMC running on it is always reported as an incomplete host, since there is no information regarding the interface MAC addresses and the default UNIX command does not work in this environment.

CMD Output Attribute	CI Name	CI Attribute
constant AIX	Unix	Host Operating System
Hostname	Unix	Host Name
Hostname	Unix	Name
Domain	Unix	OS Domain Name
Ipv4addr_eth<0..N>	IpAddress	Ip Address

lspartition -c <TYPE>_<VERSION> -i**Output**

```
2,192.168.80.52,3;1,192.168.80.62,3;3,192.168.80.53,3
```

Mapping

Each block in the output is separated by the semicolon character (;). The first value is the LPAR ID and the second value is the LPAR IP address. By matching the ID of the LPAR with output from other commands an incomplete host is created and reported with an assigned LPAR Profile CI.

lssyscfg -r sys

Output

```
name=XXXXXXXX-XXXX-XXX-XXXXXXXXXX-XX,type_model=XXXX-XXX,
serial_num=XXXXXX,ipaddr=192.168.1.10,state=Operating,sys_time=04/27/2010
12:55:23,power_off_policy=1,active_lpar_mobility_capable=0,inactive_lpar_mobility_ca
pable=0,active_lpar_share_idle_procs_capable=0,active_mem_sharing_capable=0,bsr
_capable=0,cod_mem_capable=0,cod_proc_capable=1,electronic_err_reporting_capa
ble=0,firmware_power_saver_capable=0,hardware_power_saver_capable=0,hardware
_discovery_capable=0,addr_broadcast_perf_policy_capable=0,hca_capable=1,huge_p
age_mem_capable=1,lhea_capable=0,lpar_avail_priority_capable=0,lpar_proc_compat
_mode_capable=0,micro_lpar_capable=1,os400_capable=0,5250_application_capable
=0,redundant_err_path_reporting_capable=1,shared_eth_failover_capable=1,sni_msg
_passing_capable=0,sp_failover_capable=1,vet_activation_capable=1,virtual_fc_capa
ble=0,virtual_io_server_capable=1,virtual_switch_capable=0,assign_5250_cpw_perce
nt=0,max_lpars=40,max_power_ctrl_lpars=1,hca_bandwidth_capabilities=null,service_
lpar_id=none,curr_sys_keylock=norm,pend_sys_keylock=norm,curr_power_on_side=t
emp,pend_power_on_side=temp,curr_power_on_speed=fast,pend_power_on_speed=
fast,curr_power_on_speed_override=none,pend_power_on_speed_override=none,po
wer_on_type=power
on,power_on_option=standby,power_on_lpar_start_policy=userinit,pend_power_on_op
tion=standby,pend_power_on_lpar_start_policy=userinit,power_on_method=02,power_
on_attr=0000,sp_boot_attr=0000,sp_boot_major_type=08,sp_boot_minor_type=01,sp_
version=00030030,mfg_default_config=0,curr_mfg_default_ipl_source=a,pend_mfg_de
fault_ipl_source=a,curr_mfg_default_boot_mode=norm,pend_mfg_default_boot_mode
=norm
```

Mapping

For each detected IBM Pseries Frame, a Hypervisor CI is created with the set name attribute IBM Hypervisor.

The output of this command is used to fill in the attributes of the **IBM PSeries Frame CI**:

CMD Output Attribute	CI Name	CI Attribute
Name	IBM PSeries Frame	Name
serial_number	IBM PSeries Frame	Host Key
cod_proc_capable	IBM PSeries Frame	CPU Capacity on Demand Capable
cod_mem_capable	IBM PSeries Frame	Memory Capacity on Demand Capable
huge_page_mem_capable	IBM PSeries Frame	Huge Memory Page Capable
max_lpars	IBM PSeries Frame	Max LPARs
Status	IBM PSeries Frame	Frame State
micro_lpar_capable	IBM PSeries Frame	Micro LPAR Capable
service_lpar_id	IBM PSeries Frame	Service LPAR ID
service_lpar_name	IBM PSeries Frame	Service LPAR Name

lshwres -r proc --level sys -m '<Managed System Name>'

Output

```
configurable_sys_proc_units=4.0,curr_avail_sys_proc_units=1.4,
pend_avail_sys_proc_units=1.4,installed_sys_proc_units=4.0,
max_capacity_sys_proc_units=deprecated,deconfig_sys_proc_units=0,
min_proc_units_per_virtual_proc=0.1,max_virtual_procs_per_lpar=64,max_procs_per_
lpar=4,max_curr_virtual_procs_per_aixlinux_lpar=64,max_curr_virtual_procs_per_vios_
_lpar=64,
max_curr_virtual_procs_per_os400_lpar=64,max_curr_procs_per_aixlinux_lpar=4,
max_curr_procs_per_vios_lpar=4,max_curr_procs_per_os400_lpar=4,
max_shared_proc_pools=1
```


Mapping

The output of this command is used to fill in the attributes of the **IBM PSeries Frame CI**:

CMD Output Attribute	CI Name	CI Attribute
min_proc_units_per_virtual_proc	IBM PSeries Frame	Min CPU Units per Virtual CPU
curr_avail_sys_proc_units	IBM PSeries Frame	Current Available CPU Units
max_shared_proc_pools	IBM PSeries Frame	Max Shared CPU Pools
configurable_sys_proc_units	IBM PSeries Frame	Configurable CPU Units
installed_sys_proc_units	IBM PSeries Frame	Installed CPU Units
pend_avail_sys_proc_units	IBM PSeries Frame	Pending Available CPU Units
max_procs_per_lpar	IBM PSeries Frame	Max CPUs per LPAR
max_virtual_procs_per_lpar	IBM PSeries Frame	Max Virtual CPUs per LPAR

lshwres -r mem --level sys -m '<Managed System Name>'

Output

```
configurable_sys_mem=32768,curr_avail_sys_mem=1344,pend_avail_sys_mem=1344,
installed_sys_mem=32768,max_capacity_sys_mem=deprecated,deconfig_sys_mem=0,
sys_firmware_mem=704,mem_region_size=64,configurable_num_sys_huge_pages=0,
curr_avail_num_sys_huge_pages=0,pend_avail_num_sys_huge_pages=0,
max_num_sys_huge_pages=1,requested_num_sys_huge_pages=0,huge_page_size=16384,
max_mem_pools=0
```

Mapping

The output of this command is used to fill in the attributes of the **IBM PSeries Frame CI**:

CMD Output Attribute	CI Name	CI Attribute
configurable_sys_mem	IBM PSeries Frame	Configurable System Memory
max_num_sys_huge_pages	IBM PSeries Frame	Max Number of Huge Pages
huge_page_size	IBM PSeries Frame	Huge Page Size
sys_firmware_mem	IBM PSeries Frame	Firmware Memory
mem_region_size	IBM PSeries Frame	Memory Region Size
curr_avail_sys_mem	IBM PSeries Frame	Current Available Memory
installed_sys_mem	IBM PSeries Frame	Installed Memory
requested_num_sys_huge_pages	IBM PSeries Frame	Requested Number of Huge Pages
pend_avail_sys_mem	IBM PSeries Frame	Pending Available Memory

lshwres -r proc --level pool -m '<Managed System Name>'

Output

```
configurable_pool_proc_units=4.0,curr_avail_pool_proc_units=1.4,pend_avail_pool_proc_units=1.4
```

Mapping

If there are no user-defined pools, the **pool_id** parameter does not appear in the output (**pool_id** is considered by the system to be zero by default).

The output of this command is used to fill in the attributes of the **IBM Processor Pool CI**:

CMD Output Attribute	CI Name	CI Attribute
curr_avail_pool_proc_units	IBM Processor Pool	CPU Pool Available Physical CPUs
configurable_pool_proc_units	IBM Processor Pool	CPU Pool Configurable Physical CPUs
pend_avail_pool_proc_units	IBM Processor Pool	CPU Pool Pending Available Physical CPUs
pool_id	IBM Processor Pool	Name

Issyscfg -r lpar -m '<Managed System Name>'

Output

```
name=somelparname1,lpar_id=5,lpar_env=aixlinux,state=Running,resource_config=1,
os_version=Unknown,logical_serial_num=65B922G5,default_profile=somedefaultprofil
ename1,curr_profile=somelparprofilename1,work_group_id=none,shared_proc_pool_u
til_auth=1,allow_perf_collection=1,power_ctrl_lpar_ids=none,boot_mode=sms,lpar_key
lock=norm,auto_start=0,redundant_err_path_reporting=0
```

Mapping

The output of this command is used to fill in the attributes of the **IBM LPAR Profile CI**:

CMD Output Attribute	CI Name	CI Attribute
logical_serial_num	IBM LPAR Profile	LPAR Serial Number
boot_mode	IBM LPAR Profile	LPAR Profile Boot Mode
auto_start	IBM LPAR Profile	LPAR Profile Auto Start
work_group_id	IBM LPAR Profile	LPAR Profile Workgroup ID
default_profile	IBM LPAR Profile	LPAR default profile name
curr_profile	IBM LPAR Profile	LPAR profile name

CMD Output Attribute	CI Name	CI Attribute
power_ctrl_lpar_ids	IBM LPAR Profile	LPAR power control ids
State	IBM LPAR Profile	Lpar state
lpar_env	IBM LPAR Profile	Lpar type
lpar_id	IBM LPAR Profile	LPAR ID
Name	IBM LPAR Profile	LPAR Name

Issyscfg -r prof -m '<Managed System Name>'

Output

```
name=name1,lpar_name=name2,lpar_id=5,lpar_env=aixlinux,all_resources=0,min_mem=4096,desired_mem=8192,max_mem=8192,min_num_huge_pages=0,desired_num_huge_pages=0,max_num_huge_pages=0,proc_mode=shared,min_proc_units=0.3,desired_proc_units=0.5,max_proc_units=1.0,min_procs=1,desired_procs=2,max_procs=2,sharing_mode=uncap,uncap_weight=128,io_slots=none,lpar_io_pool_ids=none,max_virtual_slots=10,"virtual_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1",
"virtual_scsi_adapters=5/client/1/11s12vio1/13/1,6/client/1/11s12vio1/14/1,7/client/1/11s12vio1/15/1",virtual_eth_adapters=2/0/1//0/1,hca_adapters=none,boot_mode=norm,conn_monitoring=1,auto_start=0,power_ctrl_lpar_ids=none,work_group_id=none,redundant_err_path_reporting=0
name=name3,lpar_name=name4,lpar_id=4,lpar_env=aixlinux,all_resources=0,min_mem=4096,desired_mem=10240,max_mem=10240,min_num_huge_pages=0,desired_num_huge_pages=0,max_num_huge_pages=0,proc_mode=shared,min_proc_units=0.3,desired_proc_units=0.7,max_proc_units=1.0,min_procs=1,desired_procs=2,max_procs=2,sharing_mode=uncap,uncap_weight=128,io_slots=none,lpar_io_pool_ids=none,max_virtual_slots=10,"virtual_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1",
"virtual_scsi_adapters=5/client/1/11s12vio1/10/1,6/client/1/11s12vio1/11/1,7/client/1/11s12vio1/12/1",virtual_eth_adapters=2/0/2//0/1,hca_adapters=none,boot_mode=norm,conn_monitoring=1,auto_start=0,power_ctrl_lpar_ids=none,work_group_id=none,redundant_err_path_reporting=0
```

Mapping

The output of this command is used to fill in the attributes of the **IBM LPAR Profile CI**:

CMD Output Attribute	CI Name	CI Attribute
sharing_mode	IBM LPAR Profile	LPAR Profile Sharing Mode
proc_mode	IBM LPAR Profile	LPAR Profile CPU Mode
uncap_weight	IBM LPAR Profile	LPAR Profile Uncapped Weight
desired_num_huge_pages	IBM LPAR Profile	LPAR Profile Desired Number of Huge Memory Pages
min_num_huge_pages	IBM LPAR Profile	LPAR Profile Minimum Number of Huge Memory Pages
max_procs	IBM LPAR Profile	LPAR Profile Maximum Number of CPUs
desired_procs	IBM LPAR Profile	LPAR Profile Desired Number of CPUs
min_proc_units	IBM LPAR Profile	LPAR Profile Minimum Physical CPUs
max_mem	IBM LPAR Profile	LPAR Profile Maximum memory
conn_monitoring	IBM LPAR Profile	LPAR Profile Connection Monitoring Enabled
min_mem	IBM LPAR Profile	LPAR Profile Minimum Memory on this LPAR
max_virtual_slots	IBM LPAR Profile	LPAR Profile Maximum Number of Virtual Slots
redundant_err_path_reporting	IBM LPAR Profile	LPAR Profile Redundant Error Path Reporting
max_num_huge_pages	IBM LPAR Profile	LPAR Profile Maximum Number of Huge Memory Pages

CMD Output Attribute	CI Name	CI Attribute
min_procs	IBM LPAR Profile	LPAR Profile Minimum Number of CPUs
max_proc_units	IBM LPAR Profile	LPAR Profile Maximum Physical CPUs
io_slots	IBM LPAR Profile	LPAR Profile IO Slots
lpar_io_pool_ids	IBM LPAR Profile	LPAR Profile IO Pool IDs
desired_proc_units	IBM LPAR Profile	LPAR Profile Desired Physical CPUs
desired_mem	IBM LPAR Profile	LPAR Profile Memory Requested by this LPAR
virtual_serial_adapters	IBM LPAR Profile	LPAR Profile Virtual Serial Adapters

lshwres -r virtualio --subtype eth --level lpar -m '<Managed System Name>'

Output

```
lpar_name=name1,lpar_id=1,slot_num=2,state=1,is_required=1,is_trunk=1,trunk_prio
rity=1,ieee_virtual_eth=0,port_vlan_id=1,addl_vlan_ids=,mac_addr=765920001002
lpar_name=l11s12vio1,lpar_id=1,slot_num=3,state=1,is_required=1,is_trunk=1,trunk_p
riority=1,ieee_virtual_eth=0,port_vlan_id=2,addl_vlan_ids=,mac_addr=765920001003
lpar_name=name2,lpar_id=2,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual
_eth=0, port_vlan_id=1,addl_vlan_ids=,mac_addr=765920002002
lpar_name=name3,lpar_id=3,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual
_eth=0, port_vlan_id=1,addl_vlan_ids=,mac_addr=765920003002
lpar_name=name4,lpar_id=4,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual
_eth=0, port_vlan_id=2,addl_vlan_ids=,mac_addr=765920004002
lpar_name=name5,lpar_id=5,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual
_eth=0, port_vlan_id=1,addl_vlan_ids=,mac_addr=765920005002
```

Mapping

The `mac_addr` attribute is represented in the Dec form without leading zeros. This value is transformed to the Hex value and left padded with missing zeros, to assure a proper representation of the MAC address in the CMDB.

Based on the MAC address, the virtual NICs are created and attached to the corresponding LPAR or VIO server, and are described by **Lpar_name** or **Lpar_id**. The **Vlan** CI is created based on **vlan_id** or **addl_vlan_ids** and is linked to the ports of the interfaces. The root container for the VLAN is a specific IBM PSeries Frame (Managed System).

CMD Output Attribute	CI Name	CI Attribute
port_vlan_id/addl_vlan_ids	VLAN	Vlan Number
IBM PSeries Frame CMDB ID	VLAN	Root Container
mac_addr (converted to Hex if needed and normalized)	Interface	MAC Address

lshwres -r virtualio --rsubtype scsi -m '<Managed System Name>'

Output

```
lpar_name=vioname1,lpar_id=1,slot_num=15,state=1,is_required=0,adapter_type=server,remote_lpar_id=5,remote_lpar_name=lpaname1,remote_slot_num=7
lpar_name=vioname1,lpar_id=1,slot_num=14,state=1,is_required=0,adapter_type=server,remote_lpar_id=5,remote_lpar_name=lpaname2,remote_slot_num=6
lpar_name=vioname1,lpar_id=1,slot_num=13,state=1,is_required=0,adapter_type=server,remote_lpar_id=5,remote_lpar_name=lpaname2,remote_slot_num=5
```

Mapping

The `lpar_name` and `lpar_id` attributes are always the name and ID of the VIO server that creates and grants the Virtual SCSI to the LPARs. The SCSI Adapter on the LPAR is identified by its slot number and the LPAR name it belongs to.

CMD Output Attribute	CI Name	CI Attribute
Slot_num/remote_slot_num	SCSI	Slot Number
Host ID with name <lpar_name> or <Remote LPAR Name>	SCSI	Root Container

lshwres -r proc --level lpar -m '<Managed System Name>'

Output

```
lpar_name=name1,lpar_id=5,curr_shared_proc_pool_id=0,curr_proc_mode=shared,curr_min_proc_units=0.3,curr_proc_units=0.5,curr_max_proc_units=1.0,curr_min_procs=1,curr_procs=2,curr_max_procs=2,curr_sharing_mode=uncap,curr_uncap_weight=128,pend_shared_proc_pool_id=0,pend_proc_mode=shared,pend_min_proc_units=0.3,pend_proc_units=0.5,pend_max_proc_units=1.0,pend_min_procs=1,pend_procs=2,pend_max_procs=2,pend_sharing_mode=uncap,pend_uncap_weight=128,run_proc_units=0.5,run_procs=2,run_uncap_weight=128
```

Mapping

Using the "`lpar_name`"/"`lpar_id`" along with the "`curr_shared_proc_pool_id`" from the output we can create corresponding links to the particular Shared Processor Pool ("IBM Processor Pool") the LPar uses. In case of the dedicated ("ded") CPU we will create links to the spare processors.

lshwres -r io --rsubtype slot -m '<Managed System Name>'**Output**

```
unit_phys_loc=XXXXX.XXX.XXXXXXX,bus_id=2,phys_loc=C3,drc_index=21010002,lp
ar_name=name1,lpar_id=1,slot_io_pool_id=none,description=RAID
Controller,feature_codes=none,pci_vendor_id=1069,pci_device_id=B166,pci_subs_ve
ndor_id=1014,pci_subs_device_id=0278,pci_class=0104,pci_revision_id=04,bus_grou
ping=0,iop=0,parent_slot_drc_index=none,drc_name=XXXXX.XXX.XXXXXXX-XX-XX
```

Mapping

The output of this command is used to create the **I/O Slot** CI. Using the name and ID of the LPAR, discovery creates the relationship to the particular LPAR that is using the slot.

CMD Output Attribute	CI Name	CI Attribute
Description	I/O Slot	Name of the Slot
bus_id	I/O Slot	Slot Bus ID
phys_loc	I/O Slot	Slot Physical Location on Bus
pci_revision_id	I/O Slot	Slot PCI Revision ID
bus_grouping	I/O Slot	Slot Bus Grouping
pci_device_id	I/O Slot	Slot PCI Device ID
unit_phys_loc	I/O Slot	Slot Physical Location
parent_slot_drc_index	I/O Slot	Slot Parent Slot DRC Index
drc_index	I/O Slot	Slot DRC Index
pci_subs_vendor_id	I/O Slot	Slot PCI Subslot Vendor ID
pci_class	I/O Slot	Slot PCI Class
slot_io_pool_id	I/O Slot	Slot IO Pool ID
pci_vendor_id	I/O Slot	Slot PCI Vendor ID
drc_name	I/O Slot	Slot DRC Name

CMD Output Attribute	CI Name	CI Attribute
feature_codes	I/O Slot	Slot Feature Codes
pci_subs_device_id	I/O Slot	Slot PCI Subslot Device ID

VIO Server Side Commands

This section includes the following commands:

- `"/usr/ios/cli/ioscli lsdev -dev 'ent*' -field name physloc -fmt"` on page 499
- `"ioscli entstat -all '<Interface Name>' | grep -E "ETHERNET STATISTICS|Device Type|Hardware Address"` on page 500
- `"ioscli entstat -all 'ent16' | grep -E "ETHERNET STATISTICS|Device Type|Hardware Address"` on page 500
- `"ioscli lsdev -dev '<Interface Name>' -attr"` on page 501
- `"ioscli lsdev -dev 'ent16' -attr"` on page 501
- `"ioscli lsmmap -all -net"` on page 502
- `"ioscli lsdev -dev fcs* -field name physloc description -fmt"` on page 502
- `"lspv"` on page 503
- `"lsvg"` on page 503
- `"lsvg <Volume Group Name>"` on page 504
- `"lsvg -lv <Volume Group Name>"` on page 505
- `"lsvg -pv <Logical Volume Group>"` on page 506
- `"lslv <Logical Volume Name>"` on page 507
- `"ioscli lsmmap -all"` on page 508

```
/usr/ios/cli/ioscli lsdev -dev 'ent*' -field name physloc  
-fmt
```

Output

```
ent0: U100C.001.DQDE777-P1-C4-T1  
ent1:U100C.001.DQDE777-P1-C4-T2  
ent2:U100C.001.DQDE777-P1-C4-T3  
ent16:  
ent17:  
ent18:  
ent19:  
ent20:
```

Mapping

The interface names and physical location of the particular interface are the output of this command. The output is split at the colon character (:) line by line; the first part is the interface name and the last is the physical location. A physical location is not always present, for example, it is not set for the SEA and Link Aggregation Interface. The physical location value is used to create a link from the physical NIC to the I/O slot.

```
ioscli entstat -all '<Interface Name>' | grep -E "ETHERNET  
STATISTICS|Device Type|Hardware Address
```

```
ioscli entstat -all 'ent16' | grep -E "ETHERNET  
STATISTICS|Device Type|Hardware Address
```

Output

```
ETHERNET STATISTICS (ent16) :  
Device Type: Shared Ethernet Adapter  
Hardware Address: 00:1B:64:91:74:55  
ETHERNET STATISTICS (ent14) :  
Device Type: EtherChannel  
Hardware Address: 00:1B:64:91:74:55  
ETHERNET STATISTICS (ent0) :  
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)  
Hardware Address: 00:1a:64:91:74:44  
ETHERNET STATISTICS (ent2) :  
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)  
Hardware Address: 00:1B:64:91:74:55  
ETHERNET STATISTICS (ent4) :  
Device Type: Virtual I/O Ethernet Adapter (I-lan)  
Hardware Address: 46:61:fa:d4:bf:0b
```

Mapping

UCMDB Version 8.0x: There cannot be two interfaces with the same MAC on a single machine. In this case the MAC Address attribute for the first interface only takes the value of the MAC address, while the other interfaces contain an underscore (_) and interface index. For example, for the above output interface **ent0** is reported with MAC Address set to **00:1B:64:91:74:55** while interface **ent2** is reported with MAC Address set to **00:1B:64:91:74:55_2**.

UCMDB Version 9.0x: This limitation is not relevant so the topology is reported as is.

CMD Output Attribute	CI Name	CI Attribute
ETHERNET STATISTICS line	Interface	Name
Hardware Address	Interface	Mac Address

CMD Output Attribute	CI Name	CI Attribute
Device Type	Interface	Description
ETHERNET STATISTICS line when Device Type value is EtherChannel	Interface Aggregation	Name
ETHERNET STATISTICS line when Device Type value is Shared Ethernet Adapter	IBM SEA	Name

ioscli lsdev -dev '<Interface Name>' -attr

ioscli lsdev -dev 'ent16' -attr

Output

```
attribute value description user_settable
adapter_names ent0,ent4 EtherChannel Adapters True
alt_addr 0x000000000000 Alternate EtherChannel Address True
auto_recovery yes Enable automatic recovery after failover True
backup_adapter NONE Adapter used when whole channel fails True
hash_mode default Determines how outgoing adapter is chosen True
mode standard EtherChannel mode of operation True
netaddr 0 Address to ping True
no_loss_failover yes Enable lossless failover after ping failure True
num_retries 3 Times to retry ping before failing True
retry_time 1 Wait time (in seconds) between pings True
use_alt_addr no Enable Alternate EtherChannel Address True
use_jumbo_frame no Enable Gigabit Ethernet Jumbo Frames True
```

Mapping

The adapter_names attribute value is used to create links to the back-up devices.

The value of Media Speed represents both Duplex and the connection Speed.

CMD Output Attribute	CI Name	CI Attribute
media_speed	Interface Index	Speed

ioscli lsmap -all -net**Output**

```

SVEA Physloc
-----
ent4 U1000.E4A.06FB0D1-V1-C11-T1
SEA ent16
Backing device ent14
Status Available
Physloc

SVEA Physloc
-----
ent9 U1000.E4A.06FB0D1-V1-C16-T1

SEA ent21
Backing device ent12
Status Available
Physloc U1000.001.DQD3693-P1-C7-T3

```

Mapping

This command is used to determine the relation between the interfaces and to identify their types.

CMD Output Attribute	CI Name	CI Attribute
SEA	SEA Adapter	Name
Backing Device	Link Aggregation / Interface	Name
SVEA	Interface (virtual)	Name

ioscli lsdev -dev fcs* -field name physloc description -fmt**Output**

```

fcs0:U1000.001.DQDE996-P1-C1-T1:4Gb FC PCI Express Adapter (df1000fe)
fcs1:U1000.001.DQDE996-P1-C1-T2:4Gb FC PCI Express Adapter (df1000fe)
fcs2:U1000.001.DQDE996-P1-C2-T1:4Gb FC PCI Express Adapter (df1000fe)
fcs3:U1000.001.DQDE996-P1-C2-T2:4Gb FC PCI Express Adapter (df1000fe)

```

Mapping

The output of this command represents the Fiber Channel Host Adapters on the VIO server. This output retrieves the FC Name and FC Physical Path which are used to create a link to the I/O slot on the PFrame, and an FC Interface Description.

CMD Output Attribute	CI Name	CI Attribute
First token	Fiber Channel HBA	Name
Third token	Fiber Channel HBA	Description

lspv

Output

```
NAME PVID VG STATUS
hdisk0 001fb2d15d794e0d rootvg active
hdisk1 001fb2d18f1f70c clientvg active
```

Mapping

This command retrieves the relation between the Physical Volume and the Volume Group, then a link is created from the Volume Group to the Physical Volume.

CMD Output Attribute	CI Name	CI Attribute
VG	Physical Volume	Name
VG	Fiber Channel HBA	Name

lsvg

Output

```
rootvg clientvg
```

Mapping

This command retrieves the list of all volume groups that are present on the VIO server.

lsvg <Volume Group Name>

Output

```
VOLUME GROUP: rootvg
VG IDENTIFIER: 001fb2d10005d9000000011a5d795185
VG STATE: active
PP SIZE: 256 megabyte(s)
VG PERMISSION: read/write
TOTAL PPs: 520 (133120 megabytes)
MAX LVs: 256
FREE PPs: 372 (95232 megabytes)
LVs: 13
USED PPs: 148 (37888 megabytes)
OPEN LVs: 11
QUORUM: 2 (Enabled)
TOTAL PVs: 1
VG DESCRIPTORS: 2
STALE PVs: 0
STALE PPs: 0
ACTIVE PVs: 1
AUTO ON: yes
MAX PPs per VG: 32512
MAX PPs per PV: 1016
MAX PVs: 32
LTG size (Dynamic): 256 kilobyte(s)
AUTO SYNC: no
HOT SPARE: no
BB POLICY: relocatable
```

Mapping

This command retrieves the values for the Volume Group CI attributes.

CMD Output Attribute	CI Name	CI Attribute
VOLUME GROUP	Volume Group	Name

CMD Output Attribute	CI Name	CI Attribute
STATE	Volume Group	Volume Group State
VG IDENTIFIER	Volume Group	Volume Group ID

lsvg -lv <Volume Group Name>

Output

```

rootvg:
LV NAME TYPE LPs PPs PVs LV STATE MOUNT POINT
hd5 boot 1 1 1 closed/syncd N/A
hd6 paging 2 2 1 open/syncd N/A
paging00 paging 4 4 1 open/syncd N/A
hd8 jfs2log 1 1 1 open/syncd N/A
hd4 jfs2 1 1 1 open/syncd /
hd2 jfs2 10 10 1 open/syncd /usr
hd9var jfs2 3 3 1 open/syncd /var
hd3 jfs2 10 10 1 open/syncd /tmp
hd1 jfs2 40 40 1 open/syncd /home
hd10opt jfs2 4 4 1 open/syncd /opt
lg_dumplv sysdump 4 4 1 open/syncd N/A
VMLib_LV jfs2 56 56 1 open/syncd /var/vio/VMLib
llv jfs2 12 12 1 closed/syncd /export/lbm

```

Mapping

This command retrieves the list of all Logical Volumes that are part of the particular Volume Group, as well as the mount points if any exist. This information enables the creation of a link from the Volume Group to the Logical Volume.

CMD Output Attribute	CI Name	CI Attribute
LV Name	Logical Volume	Name
Mount Point	Disk (FS)	Name
Type	Disk	Type

lsvg -pv <Logical Volume Group>

Output

```
rootvg:  
PV_NAME PV STATE TOTAL PPs FREE PPs FREE DISTRIBUTION  
hdisk0 active 520 372 103..30..31..104..104
```

Mapping

This command retrieves the list of the Physical Volumes in the Volume Group. This information enables the creation of a link between the Physical Volume and the Volume Group.

lslv <Logical Volume Name>**Output**

```

LOGICAL VOLUME: lv1
VOLUME GROUP: clientvg
LV IDENTIFIER: 000fb1d10230d9000000011b8f1f8187.1
PERMISSION: read/write
VG STATE: active/complete
LV STATE: opened/syncd
TYPE: jfs
WRITE VERIFY: off
MAX LPs: 32512
PP SIZE: 512 megabyte(s)
COPIES: 1
SCHED POLICY: parallel
LPs: 70
PPs: 70
STALE PPs: 0
BB POLICY: non-relocatable
INTER-POLICY: minimum
RELOCATABLE: yes
INTRA-POLICY: middle
UPPER BOUND: 1024
MOUNT POINT: N/A
LABEL: None
MIRROR WRITE
CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?: NO
DEVICESUBTYPE : DS_LVZ

```

Mapping

This command retrieves information about the Logical Volume parameters, which are mapped to the attributes of the Logical Volume CI.

CMD Output Attribute	CI Name	CI Attribute
LOGICAL VOLUME	Logical Volume	Name
LV IDENTIFIER	Logical Volume	Logical Volume ID

CMD Output Attribute	CI Name	CI Attribute
LV STATE	Logical Volume	Logical Volume Status
Type	Logical Volume	Logical Volume File System Type

ioscli lsmap -all

Output

```

SVSA Physloc Client Partition ID
-----
vhost0 U1000.E4A.06FB0D1-V1-C21 0x00000002

VTD vtopt0
Status Available
LUN 0x8100000000000000
Backing device /var/vio/VMLib/bootcd_rh5
Physloc

SVSA Physloc Client Partition ID
-----
vhost3 U1000.E4A.06FB0D1-V1-C31 0x00000002

VTD vtscsi0
Status Available
LUN 0x8100000000000000
Backing device os_lv1
Physloc

VTD vtscsi1
Status Available
LUN 0x8200000000000000
Backing device p01_lv1
Physloc

VTD vtscsi8
Status Available
LUN 0x8300000000000000
Backing device p01_lv2
Physloc

```

Mapping

This command retrieves the relation from the vSCSI to the exact backing device, which is usually a Volume or a Volume Group.

CMD Output Attribute	CI Name	CI Attribute
SVSA	SCSI	Name
C<Number>	SCSI	Slot Number
Backing Device	LV/PV/FS	Name

LPAR Side Commands

This section includes the following commands:

- "lscfg" on page 510

lscfg

Output

```

INSTALLED RESOURCE LISTThe following resources are installed on the machine.+/-
= Added or deleted from Resource List.* = Diagnostic support not available.
Model Architecture: chrp Model Implementation: Multiple Processor, PCI bus +
sys0 System Object+ sysplanar0 System
Planar* vio0 Virtual I/O Bus* vsa0 U1000.505.062136A-
V1-C0 LPAR Virtual Serial Adapter* vty0 U1000.505.062136A-V1-C0-L0
Asynchronous Terminal* pci2 U1000.001.AAA0757-P1 PCI Bus* pci1
U1000.001.AAA0757-P1 PCI Bus* pci0 U1000.001.AAA0757-P1
PCI Bus* pci3 U1000.001.AAA0757-P1 PCI Bus+ ent0
U1000.001.AAA0757-P1-T1 2-Port 10/100/1000 Base-TX PCI-X Adapter
(14108902)+ ent1 U1000.001.AAA0757-P1-T2 2-Port 10/100/1000 Base-
TX PCI-X Adapter (14108902)* pci4 U1000.001.AAA0757-P1 PCI Bus+
usbhc0 U1000.001.AAA0757-P1 USB Host Controller (33103500)+ usbhc1
U1000.001.AAA0757-P1 USB Host Controller (33103500)* pci5
U1000.001.AAA0757-P1 PCI Bus* ide0 U1000.001.AAA0757-P1-T10
ATA/IDE Controller Device+ cd0 U1000.001.AAA0757-P1-D3 IDE DVD-
ROM Drive* pci6 U1000.001.AAA0757-P1 PCI Bus+ sisscsia0
U1000.001.AAA0757-P1 PCI-X Dual Channel Ultra320 SCSI Adapter+ scsi0
U1000.001.AAA0757-P1-T5 PCI-X Dual Channel Ultra320 SCSI Adapter bus+
scsi1 U1000.001.AAA0757-P1-T9 PCI-X Dual Channel Ultra320 SCSI
Adapter bus+ hdisk0 U1000.001.AAA0757-P1-T9-L5-L0 16 Bit LVD SCSI Disk
Drive (146800 MB)+ hdisk1 U1000.001.AAA0757-P1-T9-L8-L0 16 Bit LVD SCSI
Disk Drive (146800 MB)+ ses0 U1000.001.AAA0757-P1-T9-L15-L0 SCSI
Enclosure Services Device+ L2cache0 L2 Cache+ mem0
Memory+ proc0 Processor

```

Troubleshooting and Limitations

- It is possible to configure the Partition Migration of an LPAR to the PFrame. This is supported only in P6, and is presently not supported by this solution.
- VIO Server on Linux OS is not supported.

35

Hyper-V

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 514

Tasks

- ▶ Discover Hyper-V on page 515

Reference

- ▶ Discovery Mechanism on page 521

Troubleshooting and Limitations on page 528

Concepts

Overview

The **Hyper-V** package discovers the Hyper-V Aware Windows server through WMI and NTCMD. It discovers resource pools, virtual switches, virtual NICs, and virtual machines.

Tasks

Discover Hyper-V

This task includes the following steps:

- "Supported Versions" on page 515
- "Prerequisites" on page 515
- "Deploy the Package" on page 516
- "The Hyper-V Topology by Shell job" on page 516
- "The Hyper-V Topology by WMI job" on page 517
- "Discovery Workflow" on page 519
- "New/Changed Entities" on page 519
- "Sample Output" on page 520

1 Supported Versions

The **Hyper-V** package supports Windows 2008 and Windows 2008 R2.

2 Prerequisites

a Set up the following credentials:

- "NTCMD Protocol"
- "WMI Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

b Verify that you can perform WMI queries in the `\\root\virtualization` namespace on the target machine, either through WMI or through the `wmic` command when connecting through a Shell protocol.

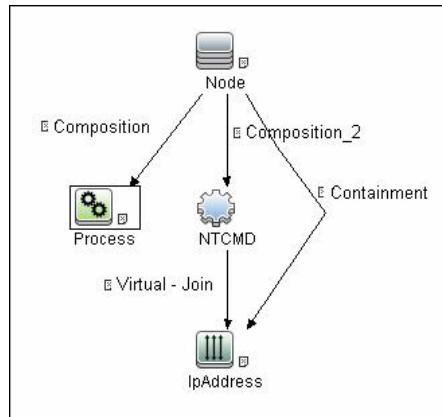
3 Deploy the Package

The name of the package is **Hyper-V**.

For details on deploying packages, see the "Package Manager" chapter in the *HP Universal CMDB Administration Guide*.

4 The Hyper-V Topology by Shell job

Trigger query:



The Process Element:

Element name: Visible Include subtypes

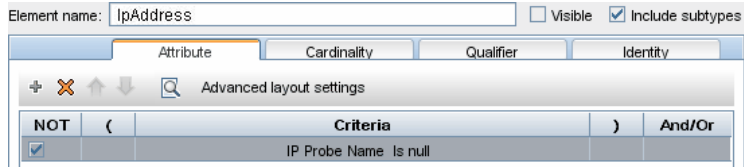
Attribute	Cardinality	Qualifier	Identity
+ X ↑ ↓ 🔍 Advanced layout settings			
NOT	(Criteria) And/Or
<input type="checkbox"/>		Name Equal ignore case "vmms.exe"	

The NTCMD Element:

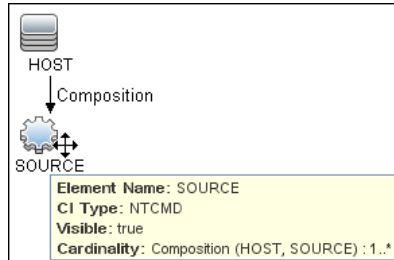
Element name: Visible Include subtypes

Attribute	Cardinality	Qualifier	Identity
+ X ↑ ↓ 🔍 Advanced layout settings			
NOT	(Criteria) And/Or
<input checked="" type="checkbox"/>		Reference to the credentials dictionary entry Is null	

The IpAddress Element:

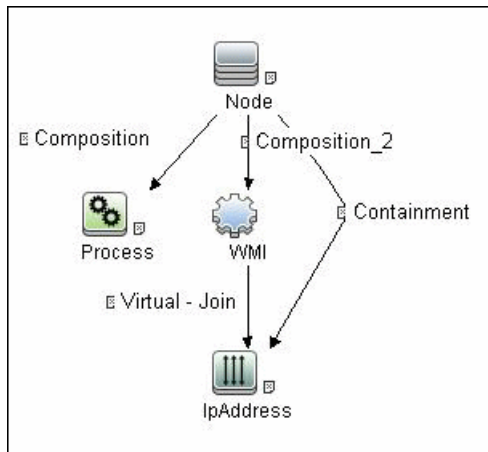


The Input Query:



5 The Hyper-V Topology by WMI job

Trigger query:



The Process Element:

Element name: Visible Include subtypes

Attribute Cardinality Qualifier Identity

+ X ↑ ↓ 🔍 Advanced layout settings

NOT	(Criteria)	And/Or
<input type="checkbox"/>		Name Equal ignore case "vmms.exe"		

The WMI Element:

Element name: Visible Include subtypes

Attribute Cardinality Qualifier Identity

+ X ↑ ↓ 🔍 Advanced layout settings

NOT	(Criteria)	And/Or
<input checked="" type="checkbox"/>		Reference to the credentials dictionary entry Is null		

The IpAddress Element:

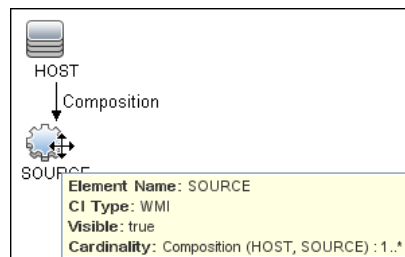
Element name: Visible Include subtypes

Attribute Cardinality Qualifier Identity

+ X ↑ ↓ 🔍 Advanced layout settings

NOT	(Criteria)	And/Or
<input checked="" type="checkbox"/>		IP Probe Name Is null		

The Input Query:



6 Discovery Workflow

To discover Hyper-V topology through Shell:

- a** Run the **Range IPs by ICMP** job to discover which of the machines in the IP range are up.
- b** Run the **Host Connection by Shell** job to discover Shell connectivity and basic information about the hosts.
- c** Run the **Host Resources and Applications by Shell** job to discover processes on target machines.
- d** Run the **Hyper-V Topology by Shell** job to discover the Hyper-V topology.

To discover Hyper-V topology through WMI:

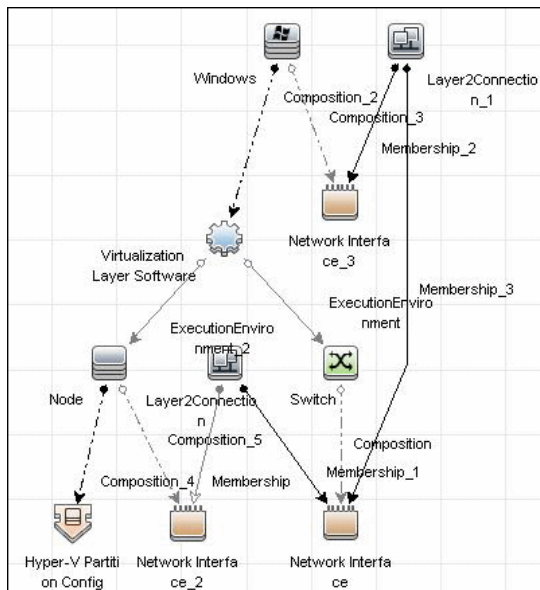
- a** Run the **Range IPs by ICMP** job to discover which of the machines in the IP range are up.
- b** Run the **Host Connection by WMI** job to discover WMI connectivity and basic information about the hosts.
- c** Run the **Host Resources and Applications by WMI** job to discover processes on target machines.
- d** Run the **Hyper-V Topology by WMI** job to discover Hyper-V topology.

7 New/Changed Entities

Entity	New/Changed	Entity Name
CITs	New	Hyper-V Partition Config (hyperv_partition_config)
Valid links	New	None
Views	New	Hyper-V Topology
Scripts	New	<ul style="list-style-type: none"> ▶ hyperv_topology_by_shell.py ▶ hyperv_topology_by_wmi.py ▶ hyperv.py
Adapters	New	<ul style="list-style-type: none"> ▶ hyperv_topology_by_shell ▶ hyperv_topology_by_wmi

Entity	New/Changed	Entity Name
Jobs	New	<ul style="list-style-type: none"> ▶ Hyper-V Topology by Shell ▶ Hyper-V Topology by WMI
Trigger Queries		<ul style="list-style-type: none"> ▶ ntcmd_on_hyperv_host ▶ wmi_on_hyperv_host
Module		Virtualization – Hyper-V (HyperV.xml)

8 Sample Output



Reference

Discovery Mechanism

This section includes the following commands:

- "Retrieve the Hyper-V Host Name" on page 522
- "Retrieve the Virtual Machine" on page 522
- "Retrieve the Global Settings for Virtual Machines" on page 522
- "Retrieve the Settings for Virtual Machines" on page 523
- "Retrieve the References from Virtual Machines to Settings (VSSD)" on page 523
- "Retrieve the References from Virtual Machine Settings (VSSD) to Components" on page 524
- "Retrieve the Memory Settings for Virtual Machines" on page 524
- "Retrieve the Processor Settings for Virtual Machines" on page 525
- "Retrieve Virtual Switches" on page 525
- "Retrieve the Ports of Virtual Switches" on page 525
- "Retrieve the References from Virtual Switches to Ports" on page 526
- "Retrieve the Interfaces of Virtual Machines" on page 526
- "Retrieve the Interfaces of Management Partitions" on page 527
- "Retrieve the References from Virtual Machines to Interfaces" on page 527
- "Retrieve the References from Ports on Virtual Switches to Interfaces" on page 527

Retrieve the Hyper-V Host Name

Object queried	Msvm_ComputerSystem
Conditions	Description = 'Microsoft Hosting Computer System'
Properties queried	ElementName
Comments	Verifies that the Hyper-V namespace \\root\virtualization is accessible and obtains the name of the Hyper-V host.

Retrieve the Virtual Machine

Object queried	Msvm_ComputerSystem
Conditions	Description = 'Microsoft Virtual Machine'
Properties queried	<ul style="list-style-type: none"> ▶ Name ▶ ElementName ▶ EnabledState ▶ HealthState
Comments	Obtains virtual machines present in the Hyper-V host, and obtains GUID, name health, and enabled states for each virtual machine.

Retrieve the Global Settings for Virtual Machines

Object queried	Msvm_VirtualSystemGlobalSettingData
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ SystemName ▶ SnapshotDataRoot ▶ ExternalDataRoot ▶ AutomaticRecoveryAction ▶ AutomaticShutdownAction ▶ AutomaticStartupAction
Comments	Obtains global settings for all virtual machines.

Retrieve the Settings for Virtual Machines

Object queried	Msvm_VirtualSystemSettingData
Conditions	None
Properties queried	<ul style="list-style-type: none"> ➤ InstanceID ➤ BaseBoardSerialNumber ➤ BIOSGUID ➤ BIOSSerialNumber ➤ ChassisAssetTag ➤ ChassisSerialNumber
Comments	<p>Obtains the VirtualSystemSettingData (VSSD) objects of the virtual machines that hold additional settings for virtual machines.</p> <p>The BIOSGUID property holds the BIOS UUID of the virtual machine. This property is stripped of leading and trailing curly brackets ({}).</p>

Retrieve the References from Virtual Machines to Settings (VSSD)

Object queried	Msvm_SettingsDefineState
Conditions	None
Properties queried	<ul style="list-style-type: none"> ➤ ManagedElement ➤ SettingData
Comments	Associates virtual machines and their settings (VirtualSystemSettingData).

Retrieve the References from Virtual Machine Settings (VSSD) to Components

Object queried	Msvm_VirtualSystemSettingDataComponent
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ GroupComponent ▶ PartComponent
Comments	Obtains references from the VirtualSystemSettingData object to its components.

Retrieve the Memory Settings for Virtual Machines

Object queried	Msvm_MemorySettingData
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ InstanceID ▶ Limit ▶ Reservation
Comments	Obtains memory settings for virtual machines (reservation and limit). The references retrieved during the previous step ("Retrieve the References from Virtual Machine Settings (VSSD) to Components" on page 524) enable the correct association of these settings to the relevant virtual machine.

Retrieve the Processor Settings for Virtual Machines

Object queried	Msvm_ProcessorSettingData
Conditions	None
Properties queried	<ul style="list-style-type: none"> ➤ InstanceID ➤ Limit ➤ Reservation ➤ Weight
Comments	Obtains processor settings for virtual machines (reservation, limit, weight). The references retrieved during a previous step ("Retrieve the References from Virtual Machine Settings (VSSD) to Components" on page 524) enable the correct association of these settings to the relevant virtual machine.

Retrieve Virtual Switches

Object queried	Msvm_VirtualSwitch
Conditions	None
Properties queried	<ul style="list-style-type: none"> ➤ ElementName ➤ Name
Comments	Obtains virtual switches configured on a Hyper-V host.

Retrieve the Ports of Virtual Switches

Object queried	Msvm_SwitchPort
Conditions	None
Properties queried	<ul style="list-style-type: none"> ➤ ElementName ➤ Name
Comments	Obtains the ports on virtual switches.

Retrieve the References from Virtual Switches to Ports

Object queried	Msvm_HostedAccessPoint
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ Antecedent ▶ Dependent
Comments	Obtains references that enable associating virtual switches and their ports.

Retrieve the Interfaces of Virtual Machines

Object queried	Msvm_VmLANEndpoint
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ Name ▶ ElementName ▶ MACAddress
Comments	Obtains endpoints that are connected to interfaces of virtual machines. Although these endpoints are not interfaces themselves, they hold enough information to report interfaces.

Retrieve the Interfaces of Management Partitions

Object queried	Msvm_SwitchLANEndpoint
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ Name ▶ ElementName ▶ MACAddress
Comments	Obtains endpoints that are connected to interfaces of a Management Partition (on a Hyper-V host). Although these endpoints are not interfaces themselves, they hold enough information to report interfaces. They include both physical interfaces and virtual interfaces of the partition used for internal connections to virtual machines.

Retrieve the References from Virtual Machines to Interfaces

Object queried	Msvm_DeviceSAPImplementation
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ Antecedent ▶ Dependent
Comments	Obtains references from virtual endpoints to virtual machines, thus enabling associations.

Retrieve the References from Ports on Virtual Switches to Interfaces

Object queried	Msvm_ActiveConnection
Conditions	None
Properties queried	<ul style="list-style-type: none"> ▶ Antecedent ▶ Dependent
Comments	Obtains references from a port on a virtual switch to endpoints that enable associations.

Troubleshooting and Limitations

Virtual machines that are offline cannot be discovered, since the information about their MAC address is not available.

36

Solaris Zones

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 530

Tasks

- ▶ Discover Solaris Zones on page 531

Reference

- ▶ The Solaris Zones_by_TTY Job on page 535
- ▶ Discovery Mechanism on page 536

Troubleshooting and Limitations on page 550

Concepts

Overview

The Solaris Zones partitioning technology is used to virtualize operating system services and provide an isolated and secure environment for running applications. A zone is a virtualized operating system environment created within a single instance of the Solaris Operating System. When you create a zone, you produce an application execution environment in which processes are isolated from the rest of the system. This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in other zones. Even a process running with superuser credentials cannot view or affect activity in other zones.

A zone also provides an abstract layer that separates applications from the physical attributes of the machine on which they are deployed. Examples of these attributes include physical device paths.

Tasks

Discover Solaris Zones

This task includes the following steps:

- "Supported Versions" on page 531
- "Prerequisites" on page 531
- "Set up Protocols" on page 531
- "Deploy the Package" on page 532
- "Discovery Workflow" on page 532
- "Created/Changed Entities" on page 532
- "Discovered CITs" on page 533
- "Sample Output" on page 534

1 Supported Versions

Solaris Zones discovery supports Solaris 10 or later.

2 Prerequisites

Zones are discovered from the Global Zone of the machine, so you should have appropriate permissions to:

- access the Global Zone and perform discovery
- log into the Non-global Zones through the **zlogin** command

3 Set up Protocols

For credentials information, see:

- "SSH Protocol"
- "Telnet Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

4 Deploy the Package

For details on deploying packages, see "Package Manager" in the *HP Universal CMDB Administration Guide*.

5 Discovery Workflow

- a Run the **Range IPs by ICMP** job to discover which of the machines in the IP range are up.
- b Run the **Host Connection by Shell** job to discover Shell connectivity and basic information about the hosts.
- c Run the **SolarisZones_by_TTY** job to discover zone configuration.

For details on running jobs, refer to "Discovery Control Panel" in the *HP Universal CMDB Data Flow Management Guide*.

6 Created/Changed Entities

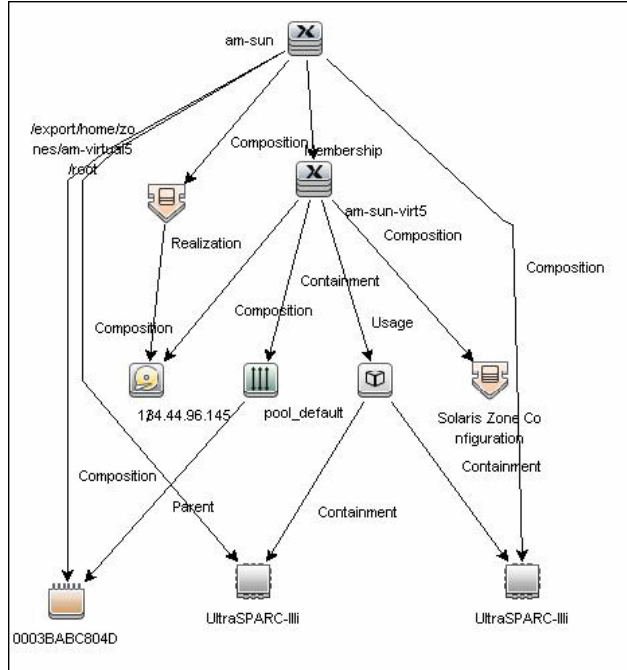
- Additional CI Types:
 - Solaris Zones Config
 - Solaris Resource Pool
- Additional valid links:
 - Solaris Resource Pool > **Containment** > CPU
 - Unix > **Usage** > Solaris Resource Pool
 - Unix > **Composition** > Solaris Resource Pool
- Modified views:
 - Solaris Zones view
- Modified scripts:
 - SolarisZone_Disc_By_TTY.py
- Additional enrichments:
 - Solaris Zones Networking

7 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

Composition
Containment
Cpu
Fibre Channel HBA
FileSystem
FileSystemExport
Interface
IpAddress
Membership
Node
Parent
Realization
Solaris Resource Pool
Solaris Zone Config
Unix
Usage

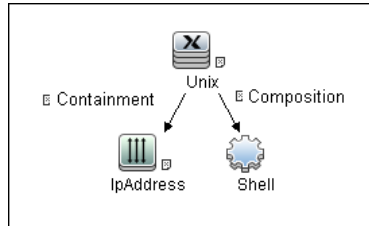
8 Sample Output



Reference

The Solaris Zones_by_TTY Job

Trigger Query



IP Process:

Element name: Visible Include subtype

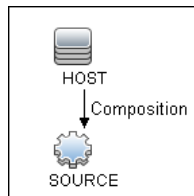
Attribute	Cardinality	Qualifier	Identity
+ X ↑ ↓ Advanced layout settings			
NOT	(Criteria) And/Or
<input checked="" type="checkbox"/>		IP Probe Name Is null	

UNIX Process:

Element name: Visible Include subtypes

Attribute	Cardinality	Qualifier	Identity
+ X ↑ ↓ Advanced layout settings			
NOT	(Criteria) And/Or
<input type="checkbox"/>		Host Operating System Like ignore case "%sunos%"	

The Input query contains one Shell CI only:



Discovery Mechanism

This section includes the following commands:

- "Verify the Connected OS is Zone-compliant" on page 536
- "Obtain List of Zones, Verify the Connected Host is Global Zone" on page 537
- "Obtain Configuration for Each of the Non-global Zones" on page 538
- "Obtain MAC Addresses for Interfaces of Global Zone" on page 541
- "Obtain IP Information for Global Zone" on page 542
- "Obtain IP Information of Exclusive Zones" on page 543
- "Obtain MAC Addresses for Dedicated Interfaces of Exclusive Zones" on page 544
- "Obtain CPU Information in Global Zone" on page 545
- "Obtain Resource Pools" on page 546
- "Obtain Fibre Channel Adapters" on page 549

Verify the Connected OS is Zone-compliant

Command	uname -r
Example of output	5.10
Values taken	5.10
Comments	This command retrieves the Solaris OS version. If it is 5.10 it is assumed that the version supports zones and discovery continues. If it is not equal to 5.10 (for example, 5.9) it is assumed the host is not zone-compliant and discovery ends with the message Server does not support zones.

Obtain List of Zones, Verify the Connected Host is Global Zone

Command	<code>/usr/sbin/zoneadm list -cp</code>
Example of output 1	<pre>0:global:running:/:native:shared 27:zone1:running:/var/opt/zones/zone1:11559a59-3c6f-6a6e-a723-cc8159351247:native:excl - :zone2:configured:/var/opt/zones/zone2::native:shared</pre>
Example of output 2 (no root permissions)	<pre>0:global:running:/ 1:am-virtual6:running:/export/home/zones/am-virtual6 5:am-virtual5:running:/export/home/zones/am-virtual5 7:am-virtual3:running:/virtual/3 9:am-virtual1:running:/am-virtual/1</pre>
Values taken	<p>Name of the zone: zone1</p> <p>Status of the zone: running</p> <p>Zone path: /var/opt/zones/zone1</p>
Comments	<p>This command gives the list of zones and their configuration including names, status, and path. The following is verified:</p> <ul style="list-style-type: none"> ▶ That global is present in the output. If it is missing, the zone that discovery connected to is not global. ▶ There is at least one more non-global zone apart from the global zone. <p>If this is not true, discovery ends with the message Server does not have zones defined.</p>

Obtain Configuration for Each of the Non-global Zones

Command	/usr/sbin/zonecfg -z <zonename> info
Example of output 1	<pre> zonename: zone1 zonepath: /var/opt/zones/zone1 brand: native autoboot: true bootargs: -m verbose pool: limitpriv: default,sys_time scheduling-class: ip-type: exclusive fs: dir: /mnt/globalzone special: /var/opt/zone1-data raw not specified type: lofs options: [] net: address not specified physical: bge2 defrouter not specified device match: /dev/bge2 dedicated-cpu: ncpus: 1 importance: 1 capped-cpu: [ncpus: 1.00] </pre>

<p>Example of output 1 (cont'd)</p>	<pre>capped-memory: physical: 16G [swap: 8G] [locked: 12G]</pre>
<p>Example of output 2</p>	<pre>zonename: zone2 zonepath: /var/opt/zones/zone2 brand: native autoboot: true bootargs: -m verbose pool: limitpriv: default scheduling-class: FSS ip-type: shared fs: dir: /mnt special: /var/opt/zone2-data raw not specified type: lofs options: [] net: address: 134.44.0.100 physical: bge0 defrouter not specified device match: /dev/pts* rctl: name: zone.cpu-shares value: (priv=privileged,limit=5,action=none)</pre>

<p>Values taken</p>	<p>The following information is obtained from the output:</p> <ul style="list-style-type: none"> ▶ brand (if it is not specified it is assumed to be native) ▶ autoboot ▶ resource pool name ▶ limit privileges ▶ scheduling class ▶ ip type ▶ all mounted file systems ▶ networking information (IP and/or network interface) ▶ dedicated CPUs and their importance ▶ memory caps ▶ cpu caps ▶ cpu shares
<p>Comments</p>	<p>This command is run for each non-global zone found. Most of these properties are stored in the Solaris Zone Config CI. File systems are reported as a File System Export from global zone to non-global. The resource pool name is used to create a link to a corresponding resource pool CI.</p>

Obtain MAC Addresses for Interfaces of Global Zone

Command	/usr/bin/netstat -np
Example of output	<pre> Net to Media Table: IPv4 Device IP Address Mask Flags Phys Addr ----- bge0 134.44.0.101 255.255.255.255 o 00:15:f2:05:9e:ff bge0 134.44.1.150 255.255.255.255 o 00:15:f2:9b:2d:96 bge0 134.44.0.100 255.255.255.255 SPLA 00:14:4f:82:74:a4 bge0 134.44.98.135 255.255.255.255 o 00:1c:c0:2b:57:35 bge0 224.0.0.0 240.0.0.0 SM 01:00:5e:00:00:00 </pre>
Values taken	MAC addresses of corresponding interfaces.
Comments	<p>This command retrieves the list of all interfaces except for the dedicated interface used in exclusive zones.</p> <p>Interfaces in the global zone are shared with shared zones, so this command runs only once.</p> <p>MAC addresses and information in the zonecfg output enables the creation of shared non-global zone Host CIs.</p>

Obtain IP Information for Global Zone

Command	<code>/usr/sbin/ifconfig -a</code>
Example of output	<pre> lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTI CAST,IPv4,VIRTUAL> mtu 8232 index 1 inet 127.0.0.1 netmask ff000000 lo0:1: flags=2001000849<UP,LOOPBACK,RUNNING,MULTI CAST,IPv4,VIRTUAL> mtu 8232 index 1 zone zone2 inet 127.0.0.1 netmask ff000000 e1000g1: flags=1000843<UP,BROADCAST,RUNNING,MULTI CAST,IPv4> mtu 1500 index 2 inet 134.44.0.50 netmask fffff00 broadcast 134.44.0.255 e1000g1:1: flags=1000843<UP,BROADCAST,RUNNING,MULTI CAST,IPv4> mtu 1500 index 2 zone zone2 inet 134.44.0.100 netmask fffff00 broadcast 134.44.0.255 </pre>
Values taken	The MAC addresses of corresponding interfaces.
Comments	<p>This command retrieves the IP configuration for the global zone that is shared with corresponding shared non-global zones.</p> <p>This information is used to report IP addresses and link them to corresponding network interfaces.</p>

Obtain IP Information of Exclusive Zones

Command	<code>/usr/sbin/zlogin -l <username> <zonename></code> <code>/usr/sbin/ifconfig -a</code>
Example of output	<pre>lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTI CAST,IPv4,VIRTUAL> mtu 8232 index 1 inet 127.0.0.1 netmask ff000000 bge2: flags=201004843<UP,BROADCAST,RUNNING,MULTI CAST,DHCP,IPv4,CoS> mtu 1500 index 2 inet 134.44.0.200 netmask ffffc00 broadcast 134.44.0.255 ether 0:14:4f:82:74:a6</pre>
Values taken	All IPs that are present except loopback.
Comments	<p>This command retrieves the IP information for exclusive non-global zones. The <code>-l <user></code> switch is added to simplify setting up the sudo pattern for <code>zlogin</code>, but it can be removed from the job parameters.</p> <p>Note: Discovery runs <code>zlogin</code> for zones in a running state only.</p>

Obtain MAC Addresses for Dedicated Interfaces of Exclusive Zones

Command	<code>/usr/sbin/zlogin -l <username> <zonename></code> <code>/usr/bin/netstat -np</code>
Example of output	<pre>Net to Media Table: IPv4 Device IP Address Mask Flags Phys Addr ----- bge2 134.44.0.200 255.255.255.255 SPLA 00:14:4f:82:74:a6 bge2 224.0.0.0 240.0.0.0 SM 01:00:5e:00:00:00</pre>
Values taken	MAC addresses.
Comments	<p>MAC addresses of the interfaces are obtained together with interface names.</p> <p>Note: Discovery runs zlogin for zones in a running state only.</p>

Obtain CPU Information in Global Zone

Command	<code>/usr/sbin/psrinfo -v</code>
Example of output	<p>Status of virtual processor 0 as of: 05/03/2010 16:00:15</p> <p>on-line since 04/26/2010 19:45:40.</p> <p>The sparcv9 processor operates at 1200 MHz, and has a sparcv9 floating point processor.</p> <p>Status of virtual processor 1 as of: 05/03/2010 16:00:15</p> <p>on-line since 04/26/2010 19:45:42.</p> <p>The sparcv9 processor operates at 1200 MHz, and has a sparcv9 floating point processor.</p>
Values taken	<p>Number of virtual CPUs with IDs</p> <p>Virtual processor names (sparcv9)</p> <p>Processors speeds (1200)</p>
Comments	<p>For each instance of the virtual processor, discovery creates a CPU with a name (sparcv9) and speed (1200). They are linked to the global zone. They are also linked to the corresponding resource pool.</p>

Obtain Resource Pools

Command	/usr/sbin/pooladm
Example of output	<pre> system default string system.comment int system.version 1 boolean system.bind-default true string system.pool.objectives wt-load pool SUNWtmp_zone1 int pool.sys_id 1 boolean pool.active true boolean pool.default false int pool.importance 1 string pool.comment boolean pool.temporary true pset SUNWtmp_zone1 pool pool_default int pool.sys_id 0 boolean pool.active true boolean pool.default true int pool.importance 1 string pool.scheduler FSS string pool.comment pset pset_default </pre>

<p>Example of output (cont'd)</p>	<pre>pset SUNWtmp_zone1 int pset.sys_id 1 boolean pset.default false uint pset.min 1 uint pset.max 1 string pset.units population uint pset.load 0 uint pset.size 1 string pset.comment boolean pset.temporary true cpu int cpu.sys_id 0 string cpu.comment string cpu.status on-line</pre>
<p>Values taken</p>	<ul style="list-style-type: none"> ➤ Pools: <ul style="list-style-type: none"> ➤ Name ➤ Is default ➤ Is active ➤ Importance ➤ Scheduler ➤ Pset: <ul style="list-style-type: none"> ➤ Name ➤ Min CPUs ➤ Max CPUs ➤ Objectives <p>Relations from Pool to Pset and from Pset to assigned CPUs by IDs</p>

Comments	<p>This information enables reporting pools and links them to corresponding CPUs of the global zone by IDs. Currently discovery reports pool and its pset as one entity.</p> <p>If the resource pools facility is not used or not active discovery cannot read the configuration, but still reports the default (dummy) pool without attributes; all CPUs are linked there.</p> <p>If the non-global zone includes the name of the pool in the configuration discovery links the zone to this pool.</p> <p>If the non-global zone has a dedicated-cpu property set, discovery calculates the name of the temporary dynamic pool for linkage. The name takes the following format: SUNWtmp_<zonename>.</p>
-----------------	---

Obtain Fibre Channel Adapters

Command	/usr/sbin/fcinfo hba-port
Example of output	<pre> HBA Port WWN: 2100001c3491b18a OS Device Name: /dev/cfg/c1 Manufacturer: QLogic Corp. Model: 555-1156-02 Firmware Version: 05.01.00 FCode/BIOS Version: BIOS: 2.2; fcode: 2.1; EFI: 2.0; Serial Number: 0708R00-4259732555 Driver Name: qlc Driver Version: 20090610-3.21 Type: N-port State: online Supported Speeds: 1Gb 2Gb 4Gb Current Speed: 2Gb Node WWN: 2000001c3491b18a HBA Port WWN: 2101001c34b1b18a OS Device Name: /dev/cfg/c2 Manufacturer: QLogic Corp. Model: 555-1156-02 Firmware Version: 05.01.00 FCode/BIOS Version: BIOS: 2.2; fcode: 2.1; EFI: 2.0; Serial Number: 0708R00-4259732555 Driver Name: qlc Driver Version: 20090610-3.21 Type: N-port State: online Supported Speeds: 1Gb 2Gb 4Gb Current Speed: 2Gb Node WWN: 2001001c34b1b18a </pre>

Values taken	<ul style="list-style-type: none"> ➤ Port WWN ➤ Os Device Name ➤ Manufacturer ➤ Model ➤ Type ➤ Serial ➤ Driver version
Comments	This information enables discovery to report the Fibre Channel HBA. The OS Device Name is held by the name attribute. The Port WWN is held by the HBA WWN attribute.

Troubleshooting and Limitations

- The following warning message appears during discovery: Not enough permissions to execute command, zone is skipped.

This may indicate that the script could not retrieve network information for exclusive zones using **zlogin** due to a lack of permissions for the user performing discovery.

To solve this problem:

- Give required permissions to the user.
- Add the **zlogin** command to the list of **sudo**-enabled commands.

37

VMware

This chapter includes:

Tasks

- ▶ Discover VMware Infrastructure Topology on page 552
- ▶ Discover VMware VMotion on page 571

Reference

- ▶ **Troubleshooting and Limitations** on page 575

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 553
- "Supported VMware Servers" on page 553
- "SSL Support Details" on page 553
- "Prerequisites – Add *.jar Files" on page 554
- "Prerequisites – Run Host Discovery" on page 554
- "Prerequisites – Run WMI Discovery" on page 555
- "Prerequisites – Run Processes Discovery" on page 555
- "Prerequisites – VMware Infrastructure Permissions" on page 555
- "Network and Protocols" on page 555
- "Discovery Workflow – Overview" on page 556
- "Discovery Workflow – VMware VirtualCenter Connection by WMI and VIM" on page 557
- "Discovery Workflow – VMware VirtualCenter Topology by VIM" on page 560
- "Discovery Workflow – VMware ESX Connection by VIM" on page 564
- "Discovery Workflow – VMware ESX Topology by VIM" on page 566
- "Virtual Topology View for Clusters" on page 569
- "Virtual Topology View for Non-Clusters" on page 570
- "Licensing Topology Map" on page 571

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.

For credentials information, see:

- "WMI Protocol"
- "NTCMD Protocol"
- "SSH Protocol"
- "Telnet Protocol"

These protocols require the user name, password, and domain name (the domain name is optional for NTCmd).

- "SNMP Protocol"
- "VMware Infrastructure Management (VIM) Protocol"

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.

in *HP Universal CMDB Data Flow Management Guide*.

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:

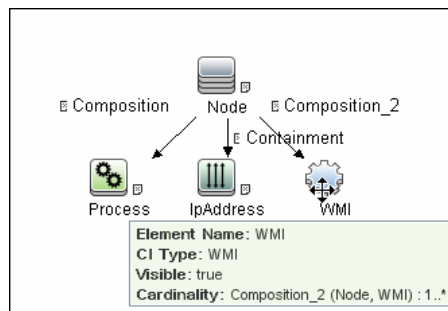
- ▶ The **Manual VMware VIM Connection** job is intended for use in those instances when the above four jobs cannot discover the VMware environment. You must, however, manually run this job, that is, you specify a URL (you need to know its format), you activate the job, and you choose the Data Flow Probe.
- ▶ 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 query:



Triggered CI Data:

- ▶ **credentialsId.** The credentials ID of the WMI agent CI.
- ▶ **ip_address.** The IP address, taken from the WMI agent CI.

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.

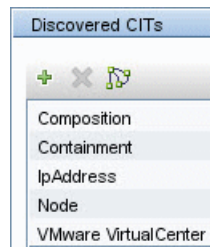
Note: DFM performs a search for the WMI port once only. The retrieved value is cached so that the same query does not need to be run for each VMware Infrastructure Management (VIM) protocol entry.

- ▶ 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:

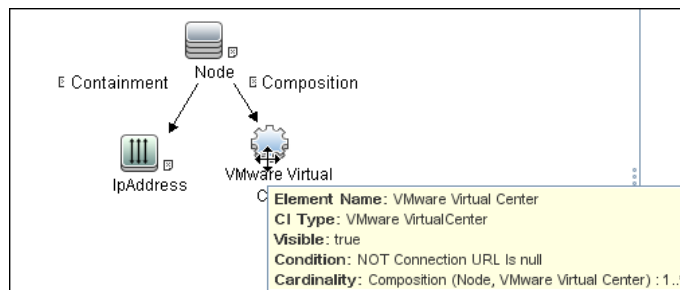
Invalid user name or password

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 query:



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.

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 561.
- 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.

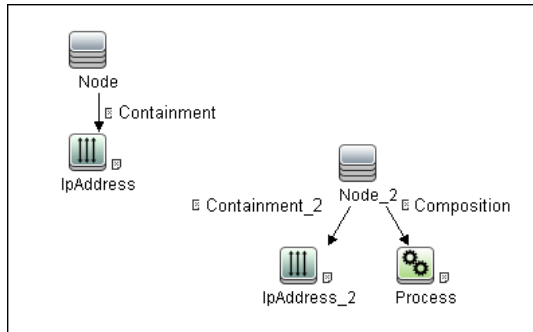
Solution. Users should have permissions for all clusters being discovered, and must have been assigned at least a Read-Only role.

13 Discovery Workflow – VMware ESX Connection by VIM

This job discovers the connections to VMware ESX servers.

Trigger CI. Unix.

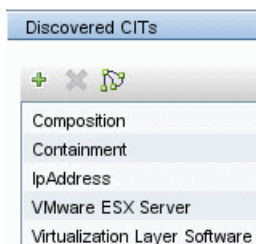
Trigger query:



Adapter parameters. None.

Data Flow Management 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
```

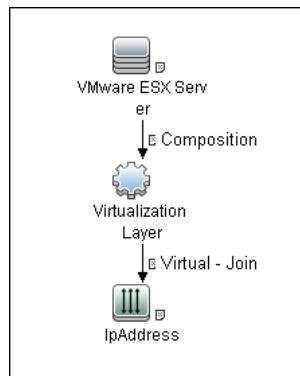
Limitation. You cannot set the connection timeout value for the job, due to VMware API limitations. The default 60 seconds timeout is always used.

14 Discovery Workflow – VMware ESX Topology by VIM

This job connects to ESX servers and discovers their topology.

Trigger CI. Virtualization Layer Software.

Trigger Query 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.

Adapter Parameters. reportPoweredOffVMs. Checks whether VMs that are powered off should be reported.

Data Flow Management performs the following procedure:

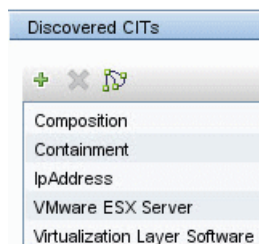
- DFM uses the connection URL (extracted from the ESX server attribute) and the user name and password (obtained by the **credentialsId** Trigger CI from the ESX server attribute) to connect to the server.

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 560.)

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>
```

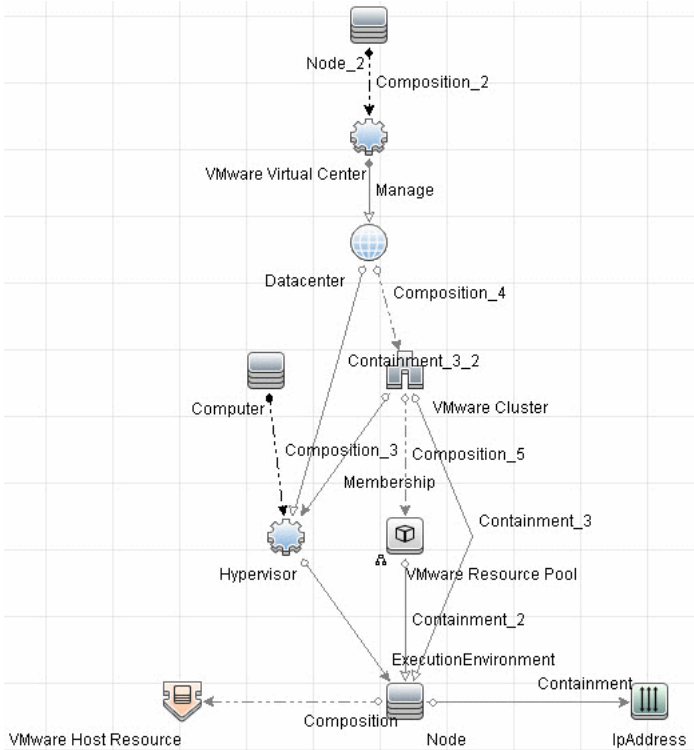
- **Problem.** The following warning message is displayed, the status is <status> and the CI is not reported:

```
Virtual machine '<vm_name>' does not have a VMware Tools running
```

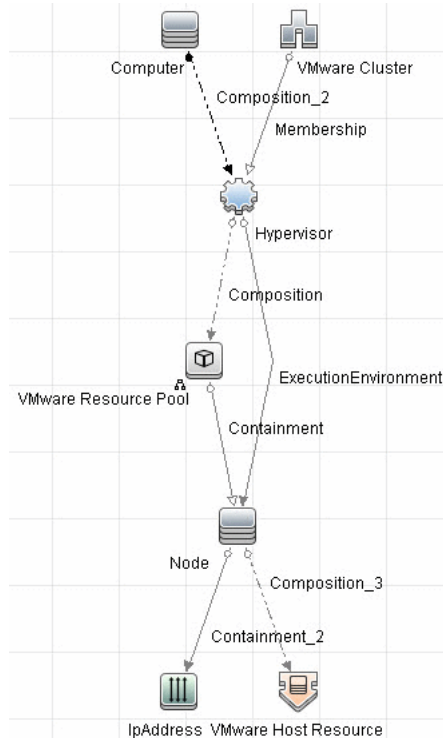
- **Problem.** The following warning message is displayed when DFM cannot retrieve license availability (permissions, in most cases, is **Global.Licenses**):

```
User does not have required '<permission>' permission, features availability information won't be reported
```

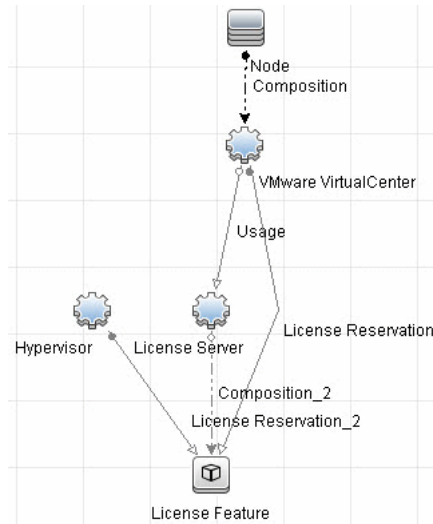

15 Virtual Topology View for Clusters



16 Virtual Topology View for Non-Clusters



17 Licensing Topology Map



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 572
- ▶ "Prerequisites" on page 572
- ▶ "Discovery Workflow" on page 573
- ▶ "DFM Package" on page 573
- ▶ "Trigger CI" on page 573
- ▶ "Trigger Query" on page 573
- ▶ "Triggered CI Data" on page 574
- ▶ "Discovered CITs" on page 574
- ▶ "Adapter Parameters" on page 574

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 *HP Universal CMDB Data Flow Management Guide*.

- 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 552.

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.

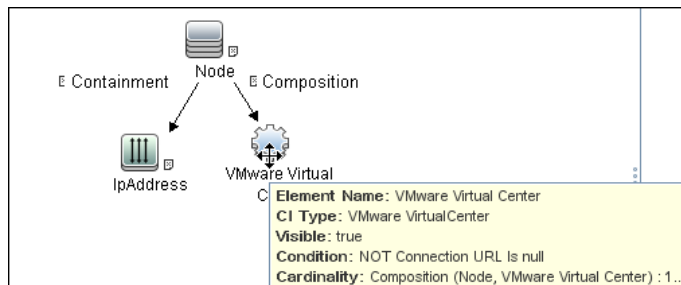
4 DFM Package

To access the **VMWare** package: **Settings > Package Manager**. For details, see "Package Manager" in the *HP Universal CMDB Administration Guide*. For details on the contents of the package, click the link to the Readme file.

5 Trigger CI

VMware VirtualCenter

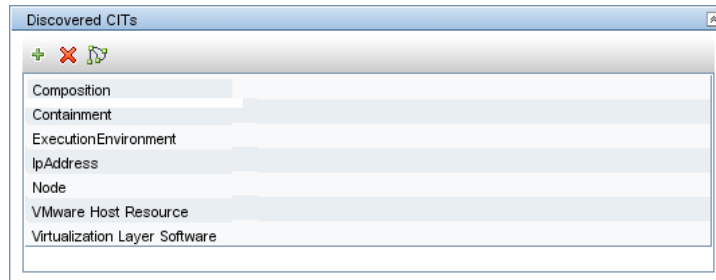
6 Trigger Query



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 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**.

Reference

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 554.

- **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`, `vsmp`, `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.

38

XEN

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 578

Tasks

- ▶ Discover Xen on page 579

Reference

- ▶ Discovery Mechanism on page 587

Concepts

Overview

The Xen hypervisor, the open source industry standard for virtualization, virtualizes x86, x86_64, IA64, ARM, and other CPU architectures. It supports guest operating systems including Windows, Linux, Solaris, and various versions of the BSD operating systems.

Tasks

Discover Xen

This task includes the following steps:

- "Supported Versions" on page 579
- "Prerequisites" on page 579
- "The Xen_by_TTY Adapter Parameters" on page 580
- "Trigger Queries" on page 581
- "Input Queries" on page 582
- "Discovery Workflow" on page 583
- "Discovered CITs" on page 583
- "Sample Output" on page 584
- "Created/Changed Entities" on page 585

1 Supported Versions

This discovery solution supports Xen 3.x or later.

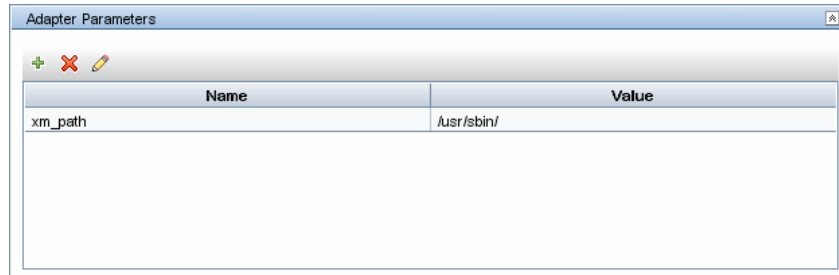
2 Prerequisites

- a** Add SSH credentials for the Xen server. For details, see "SSH Protocol" in *HP Universal CMDB Data Flow Management Guide*.
- b** If the `xm` command is not located in a standard path (for example, `/bin`, `/sbin`, `/usr/bin`, or `/usr/sbin`), you must either add the path to `xm` in the `PATH` OS environment variable, or specify the path to it in the job property.

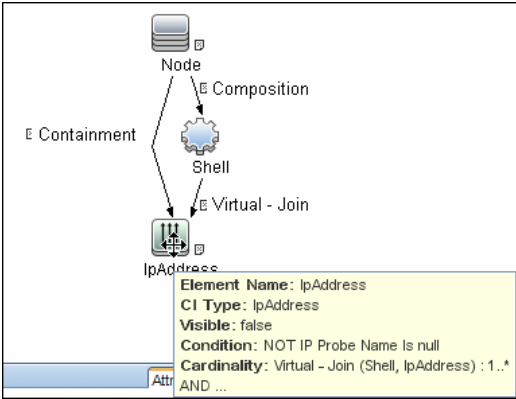
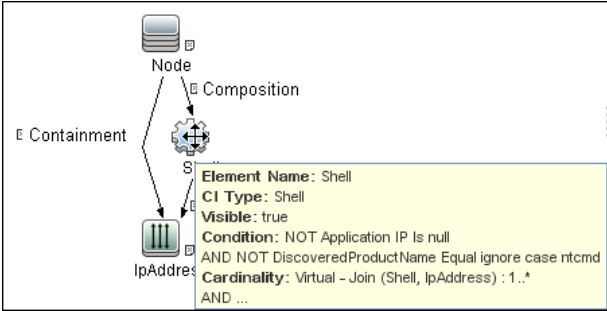
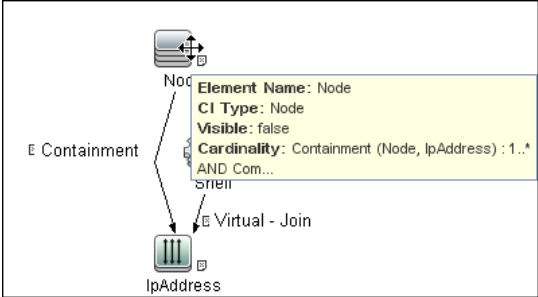
- c If some commands are configured to run with **sudo** on the target host, in the **Protocol Parameters** dialog box, fill in the following fields:
 - **Sudo paths.** Enter the full path to the **sudo** executable, together with the name of the executable. You can add more than one entry if executable files are placed in various places on the target operating systems.
Example: `sudo,/usr/bin/sudo,/bin/sudo`
 - **Sudo commands.** Enter a list of the commands that are prefixed with the **sudo**.
Example: `lspath,ifconfig`
- d Make sure that the discovery user has permissions to connect to the Xen server and to run the following commands:
 - `xm info`
 - `xm list`
 - `xm list -l <domain_name>`
 - `brctl show`
 - `ifconfig -a`

For details, see "Protocol Parameter Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

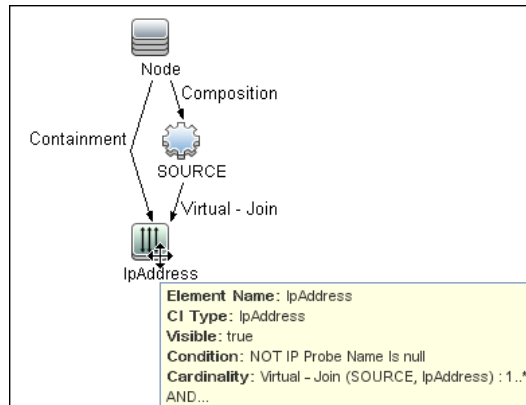
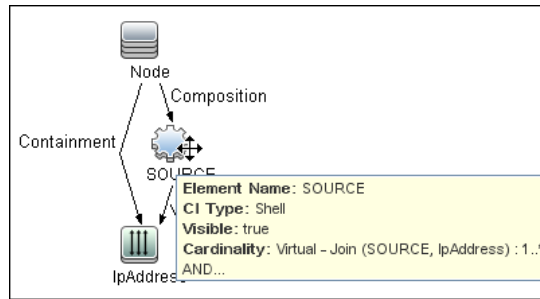
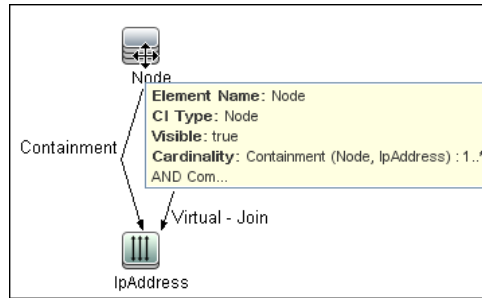
3 The Xen_by_TTY Adapter Parameters



4 Trigger Queries



5 Input Queries



Triggered CI Data:

Triggered CI Data	
Name	
Protocol	\${SOURCE.root_class}
credentialsId	\${SOURCE.credentials_id}
hostId	\${SOURCE.root_container}
ip_address	\${SOURCE.application_ip}

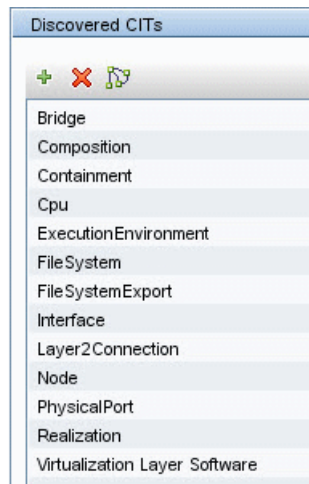
Used script: xen_by_tty.py

6 Discovery Workflow

- a Run the **Range IPs by ICMP** job.
- b Run the **Host Connection by Shell** job.
- c Run the **Xen Topology by TTY** job.

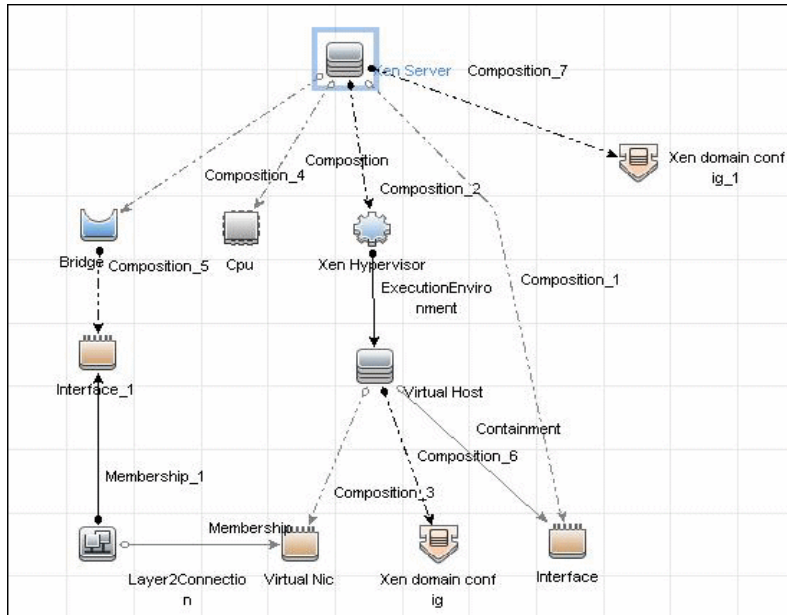
For details on running jobs, refer to the "Discovery Control Panel" chapter in *HP Universal CMDB Data Flow Management Guide*.

7 Discovered CITs

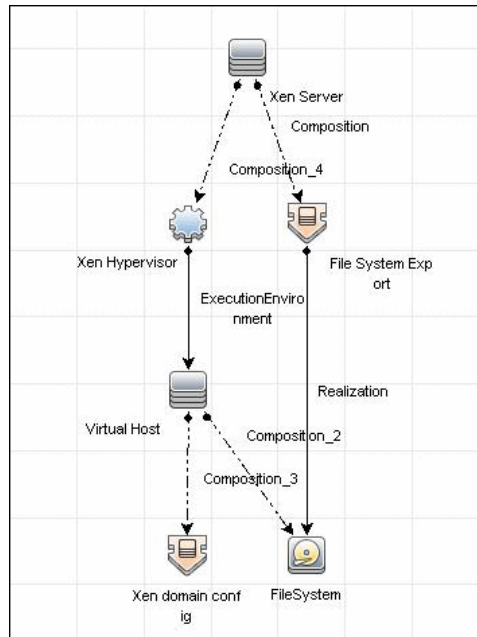


8 Sample Output

Xen Topology:



Xen Storage Topology:



9 Created/Changed Entities

Entity Name	Entity Type	Entity Description
xen_domain_config.xml	CIT	Domain configuration and parameters
Xen Topology by TTY.xml	Job	Main job
Virtualization - Xen.xml	Module	Discovery module
Xen_by_TTY.xml	Adapter	Discovery adapter
xen_by_tty.py	script	Discovery Jython script
xen_unix_with_shell.xml	query	Trigger query
Xen Topology.xml	View	View of the discovered topology

Entity Name	Entity Type	Entity Description
Xen Storage Topology.xml	View	View of the storage topology
containment.host.interface.xml	Valid link	
composition.bridge.interface.xml	Valid link	

Reference

Discovery Mechanism

This section includes the following commands:

- "Map Output to CI Attributes – for Xen Hypervisor and Hardware Resources" on page 588
- "Use Output to Create List of Domains" on page 589
- "Map Output to CI Attributes – for Domain Configuration Information" on page 589
- "Use Output to Retrieve Relationship Between Bridge and Bridged" on page 592

Map Output to CI Attributes – for Xen Hypervisor and Hardware Resources

Command	xm info
Output	<pre> host : VMAMQA348.devlab.ad release : 2.6.18-194.3.1.el5xen version : #1 SMP Sun May 2 04:26:43 EDT 2010 machine : x86_64 nr_cpus : 2 nr_nodes : 1 sockets_per_node : 2 cores_per_socket : 1 threads_per_core : 1 cpu_mhz : 2932 hw_caps : 0febfbff:28100800:00000000:00000140:80982201:00000000:0 0000001 total_memory : 8191 free_memory : 5442 node_to_cpu : node0:0-1 xen_major : 3 xen_minor : 1 xen_extra : .2-194.3.1.el5 xen_caps : xen-3.0-x86_64 xen-3.0-x86_32p xen_pagesize : 4096 platform_params : virt_start=0xffff800000000000 xen_changeset : unavailable cc_compiler : gcc version 4.1.2 20080704 (Red Hat 4.1.2-48) cc_compile_by : mockbuild cc_compile_domain : redhat.com cc_compile_date : Sun May 2 04:16:18 EDT 2010 xend_config_format : 2 </pre>

Mapping			
Output of this command is used to populate the attributes of the CIs:			
	CMD Output Attribute	CI Name	CI Attribute Display Name
	xen_major + "." xen_minor	Hypervisor	Application version (application_version_ number)
	xen_major + "." +xen_minor+ xen_extra	Hypervisor	Application Version Description
	nr_cpus	Xen domain config	Xen Number of Processors
	sockets_per_node	Xen domain config	Xen Sockets number
	threads_per_core	Xen domain config	Xen Threads per Core
	total_memory	Xen domain config	Xen Total Memory
	free_memory	Xen domain config	Xen Free Memory

Use Output to Create List of Domains

Command	xm list
Output	<pre>Name ID Mem(MiB) VCPUs State Time(s) Domain-0 0 2048 2 r---- 15771.6 fedora12_64 9 512 1 -b---- 1272.4</pre>
Mapping	The output creates a list of Domains running on the particular Xen server.

Map Output to CI Attributes – for Domain Configuration Information

Command	<pre>xm list -l fedora12_64 xm list -l <domain_name></pre>
----------------	--

Output	<pre> (domain (domid 9) (uuid d2ea72a3-7d27-933e-021e-2d7ec1f05081) (vcpus 1) (cpu_cap 0) (cpu_weight 256.0) (memory 512) (shadow_memory 0) (maxmem 512) (bootloader /usr/bin/pygrub) (features) (name fedora12_64) (on_poweroff destroy) (on_reboot restart) (on_crash restart) (image (linux (ramdisk /var/lib/xen/boot_ramdisk.pkJA8q) (kernel /var/lib/xen/boot_kernel.B7TO_v) (args 'ro root=/dev/mapper/VolGroup-lv_root LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us console=hvc0 rhgb quiet'))) (cpus ()) (device (vif (backend 0) (script vif-bridge) (bridge virbr0) (mac 00:16:36:61:12:c6))) (device (tap (backend 0) (dev xvda:disk) (uname tap:aio:/mnt/vmimages/fedora12_64.img) (mode w)))) </pre>
--------	---

Output <i>(cont'd)</i>	<pre>(state -b----) (shutdown_reason poweroff) (cpu_time 1272.36904274) (online_vcpus 1) (up_time 961277.138582) (start_time 1277970939.8) (store_mfn 2287142) (console_mfn 2287141))</pre>		
Mapping			
Output of this command is used to populate the attributes of the CIs:			
	CMD Output Attribute	CI Name	CI Attribute Display Name
	domid	Xen domain config	Xen Domain Id
	uuid	Host	host BIOS UUID
	vcpus	Xen domain config	Xen virtual CPU Count
	memory	Xen domain config	Xen Domain Memory
	name	Xen domain config	Xen Domain Name
	on_poweroff	Xen domain config	Xen Domain on Power Off Action
	on_reboot	Xen domain config	Xen Domain on Restart Action
	on_crash	Xen domain config	Xen Domain on Crash Action
	state	Xen domain config	Xen Domain State
	bridge	Bridge	Name
	uname tap:aio:	Network Share	Name
	mac	Network Interface	Interface MAC Address

Use Output to Retrieve Relationship Between Bridge and Bridged

Command	brctl show		
Output	<pre> bridge name bridge id STP enabled interfaces br0 8000.0050569f684a no eth0 peth0 virbr0 8000.fefffffffff yes vif9.0 </pre>		
Mapping			
From this output, the relationship between the bridge and bridged interfaces is retrieved.			
	CMD Output Attribute	CI Name	CI Attribute Display Name
	bridge name	Bridge	Name
	bridge id	Bridge	Bridge Base MAC Address
	interfaces	Network Interface	Name

39

Apache Tomcat

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

This chapter includes:

Concepts

- ▶ Overview on page 594

Tasks

- ▶ Discover Apache Tomcat on page 596
- ▶ Discover Bugzilla, Wordpress, and MediaWiki on page 600

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"/>
      </Cluster>
    </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 596
- "Network and Protocols" on page 596
- "Discovery Workflow" on page 597
- "Apache Tomcat CITs" on page 598
- "Apache Tomcat Links" on page 598
- "Input Query" on page 599
- "Triggered CI Data" on page 599
- "Topology View" on page 600

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 Protocol"
- "SSH Protocol"
- "Telnet Protocol"

in *HP Universal CMDB Data Flow Management Guide*.

3 Discovery Workflow

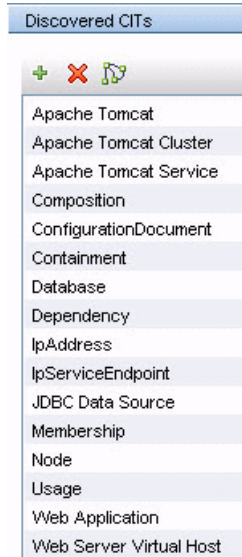
- a** Run the **Range IPs by ICMP** job (in the **Network – Basic** module) to discover IPs in the range where Tomcat is running.
- b** Run the **Host Connection by Shell** job (in the **Network – Basic** module) to discover Shell agents.
- c** Run the **Host Resources and Applications by Shell** job (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.

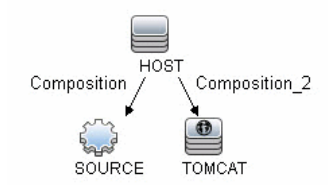
4 Apache Tomcat CITs



5 Apache Tomcat Links

- ▶ Tomcat Service > Usage > IpServiceEndpoint
- ▶ Tomcat Service > Composition > Web Server Virtual Host
- ▶ Web Application > Usage > JDBC Data Source
- ▶ Tomcat > Composition > Configuration Document
- ▶ Tomcat > Composition > Tomcat Service
- ▶ Tomcat Cluster > Membership > Web Server Virtual Host
- ▶ Web Server Virtual Host > Composition > Web Application
- ▶ Tomcat > Composition > JDBC Data Source
- ▶ JDBC Data Source > Dependency > Database

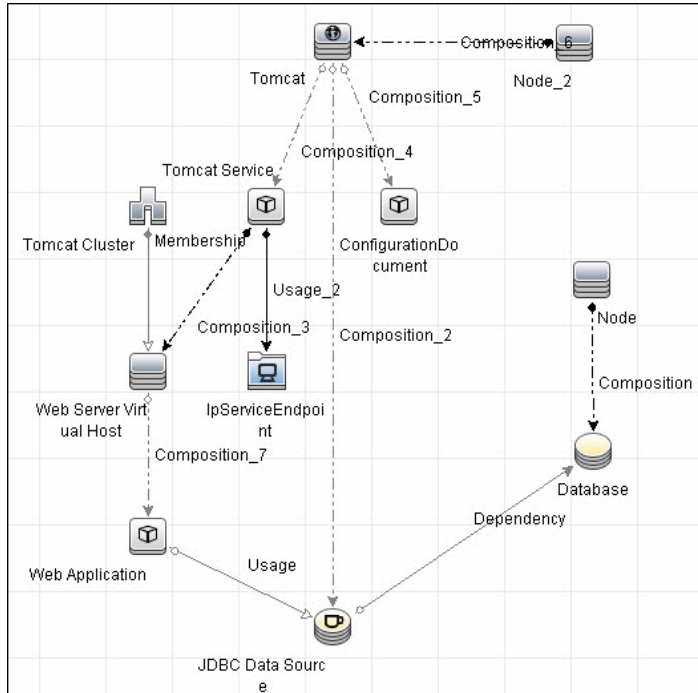
6 Input Query



7 Triggered CI Data

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

Application	Supported Version
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)

40

Microsoft Internet Information Services (IIS)

This chapter includes:

Tasks

- ▶ Discover Microsoft Internet Information Services (IIS) – Previous Topology on page 604
- ▶ Discover Microsoft Internet Information Services (IIS) – Current Topology on page 607

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 604
- "Network and Protocols" on page 604
- "Discovery Workflow" on page 604
- "Discovered CITs" on page 605
- "Topology Map" on page 606

1 Supported Versions

IIS versions 5 and 6.

2 Network and Protocols

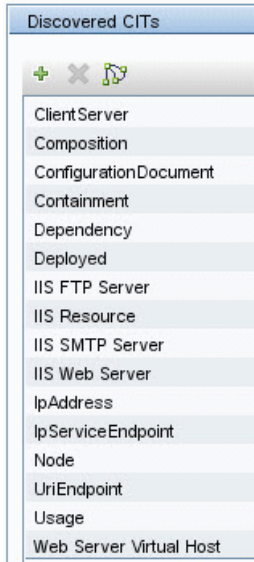
NTCmd. For credentials information, see "NTCMD Protocol" in *HP Universal CMDB Data Flow Management Guide*. Verify that the target machine running IIS lies in the Data Flow Probe range.

3 Discovery Workflow

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

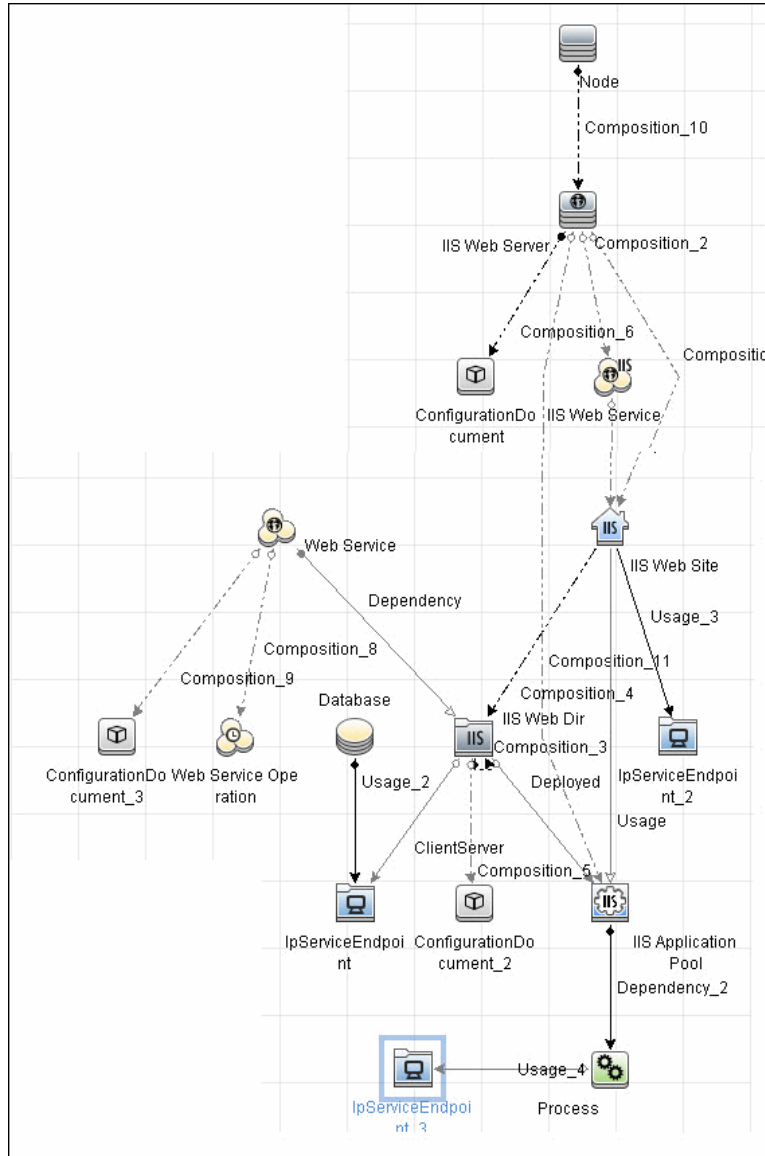
- **WebServices by URL.** For a limitation, see "Troubleshooting and Limitations" on page 725.
- **IIS Applications by NTCMD**

4 Discovered CITs



For details, see "Subgraph Condition Definition Dialog Box" in the *HP Universal CMDB Modeling Guide*.

5 Topology Map



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 607
- "Prerequisites" on page 608
- "Network and Protocols" on page 608
- "Trigger Query" on page 608
- "Triggered CI Data" on page 608
- "Adapter Parameters" on page 609
- "Discovery Workflow" on page 609
- "Discovered CITs" on page 610
- "IIS Package" on page 611
- "Permissions" on page 611
- "Topology Map" on page 612
- "Bugzilla, Wordpress, and MediaWiki Discovery" on page 612
- "Troubleshooting and Limitations" on page 613

1 Supported Versions

IIS version 7.0 or earlier.

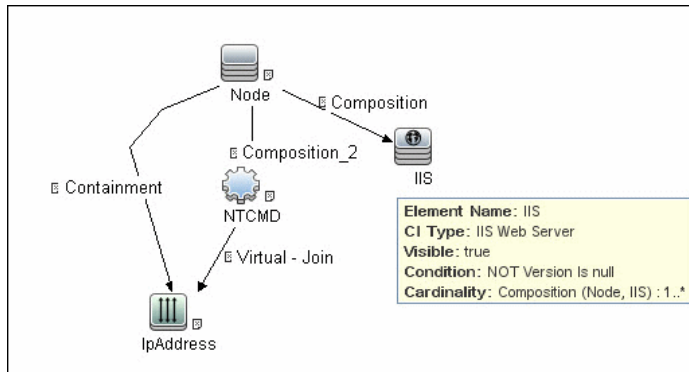
2 Prerequisites

- ▶ 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 *HP Universal CMDB Data Flow Management Guide*.

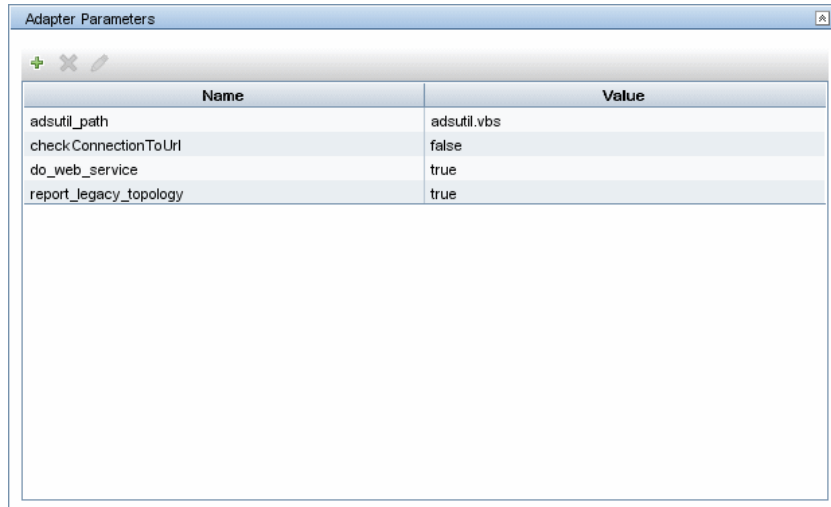
4 Trigger Query



5 Triggered CI Data

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}

6 Adapter Parameters



Name	Value
adsutil_path	adsutil.vbs
checkConnectionToUrl	false
do_web_service	true
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.

7 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.

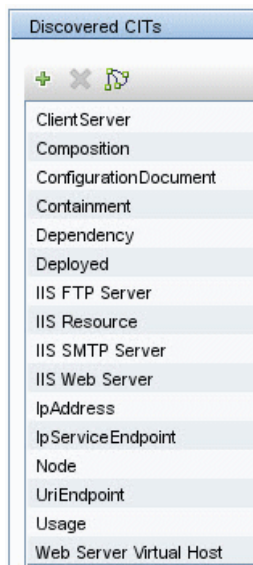
After the connection is made, DFM copies the **adsutil.vbs** script on the remote machine. DFM retrieves IIS topology information from the output of this tool.

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 611.

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.

8 Discovered CITs



9 IIS Package

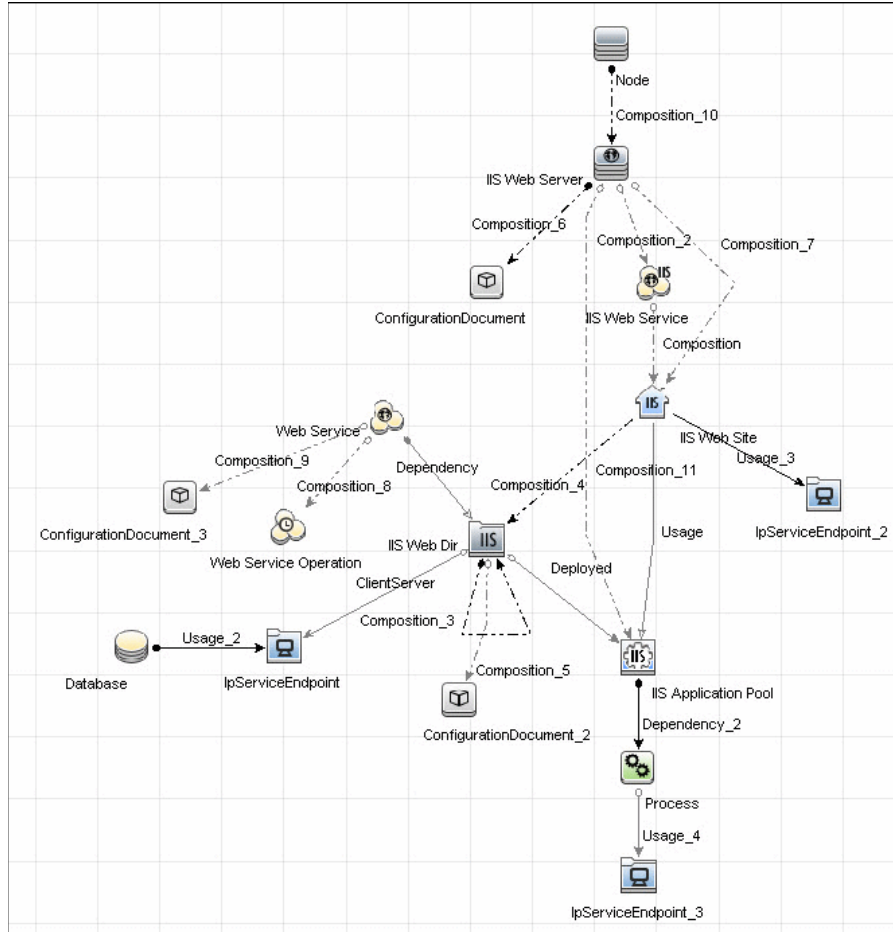
All components responsible for IIS discovery by Shell are bundled in the IIS package (in the Web Tier category).

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 600.

13 Troubleshooting and Limitations

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

41

HP ServiceCenter/Service Manager Integration

This chapter includes:

Concepts

- ▶ Adapter Usage on page 619
- ▶ Supported Versions on page 620
- ▶ Data Push Flow on page 620
- ▶ Federation Use Cases on page 621
- ▶ Viewing the Actual State on page 622
- ▶ The serviceDeskConfiguration.xml File on page 625

Tasks

- ▶ Deploy the Adapter – Typical Deployment on page 634
- ▶ Deploy the ServiceDesk Adapter on page 634
- ▶ Add an Attribute to the ServiceCenter/Service Manager CIT on page 640
- ▶ Communicate with Service Manager over SSL on page 647
- ▶ Set Up Service Manager Integration for Data Push on page 648
- ▶ Add a New Attribute to an Existing CI Type on page 651
- ▶ Add a New CI Type on page 652

Reference

- ▶ Flow and Configuration on page 654

Troubleshooting and Limitations on page 661

Note: This adapter is a specific configuration of the ServiceDesk Adapter.

Concepts

Adapter Usage

The ServiceCenter/Service Manager Adapter supports the push to and retrieval of data from HP ServiceCenter and HP Service Manager. This adapter connects to, sends data to, and receives data from ServiceCenter/Service Manager using the Web Service API. Every request to ServiceCenter/Service Manager to calculate a federated query or to push data is made through this adapter. The adapter is compatible with HP ServiceCenter version 6.2, and HP Service Manager, versions 7.0x, 7.1x, and 7.2x-9.2x (following changes to the WSDL configuration).

Data Push

The data push framework uses the adapter to push CIs and relationships to HP ServiceCenter and HP Service Manager. Once a CI has been pushed to HP ServiceCenter/HP Service Manager, an Actual State flow may be triggered in HP ServiceCenter/HP Service Manager, and selecting a tab in HP ServiceCenter/HP Service Manager enables you to view the most updated data available on the CI in UCMDB.

For details about setting up a data push flow, see "Data Push Tab" in the *HP Universal CMDB Data Flow Management Guide*.

Federation

The adapter supports three external CI types: Incident, Problem, and Planned Change. The adapter retrieves the CIs of these types from ServiceCenter/Service Manager with the required layout and by a given filter (using reconciliation and/or a CI filter). Each of these CITs can be related to one of the following UCMDB internal CITs: Host, Business Service, Application. Each UCMDB internal CIT includes a reconciliation rule in the ServiceCenter/Service Manager configuration that can be changed dynamically (for details, see "Reconciliation Data Configuration" on page 628). Note that there are no internal relationships between adapter-supported CITs.

The modeling of the supported CITs and virtual relationships is supplied with the Adapter. You can add attributes to a CIT (for details, see "Add an Attribute to the ServiceCenter/Service Manager CIT" on page 640).

For details about setting up a federation flow, see "Federation Tab" in the *HP Universal CMDB Data Flow Management Guide*.

Supported Versions

UCMDB is delivered with three different Service Manager adapters, for different versions of HP ServiceCenter/HP Service Manager. When you define an integration, choose the correct adapter according to your Service Manager version.

Data Push Flow

You can configure the data push flow options for the Service Manager integration by updating the following UCMDB, Service Manager and adapter XML files:

- ▶ **xslt files** – maps the UCMDB graph to the Service Manager request.
- ▶ **smSyncConfFile** – maps a tql name to an xslt file. This resource should be changed when adding a new TQL query.

Multi-Threading

By default, the ServiceDesk Adapter uses six concurrent threads to push data to Service Manager. To configure the ServiceDesk Adapter multi-thread settings, edit the **sm.properties** file, located in:

Data Flow Management > Adapter Management > ServiceManagerAdapter corresponding to Service Manager version > Configuration Files

Error Handling

The ServiceCenter/Service Manager adapter has a mechanism that permits the capture of CIs that failed in a push job due to specific errors, and instead of failing the entire push job, attempts to send them again in future executions. In such a case, the statistics display the **Successful with warnings** status.

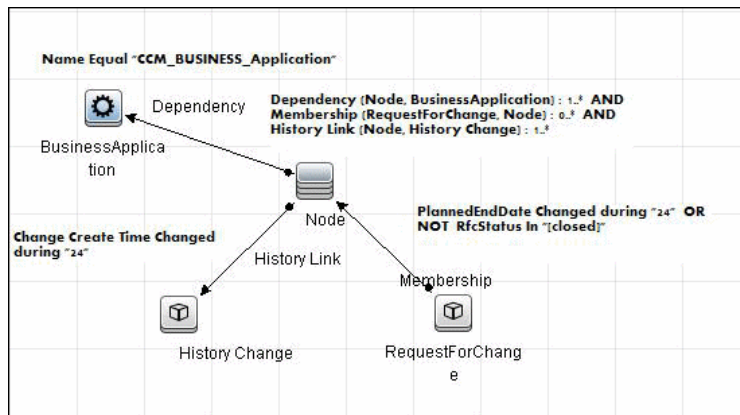
By default, only the error of locked CI (Error 3) triggers this mechanism.

To configure error handling, navigate to **Adapter Management > ServiceManagerAdapterX-X > Configuration Files > sm.properties** and set the required values.

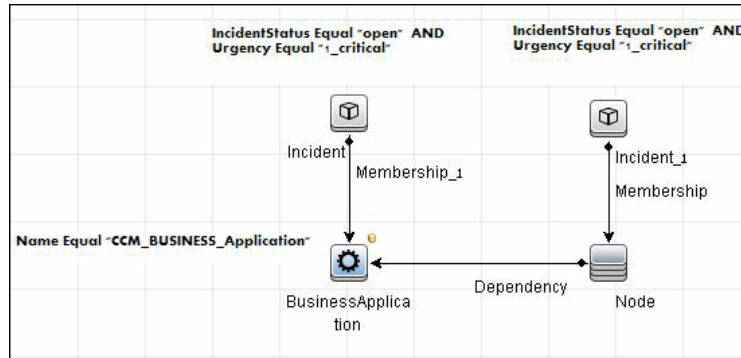
Federation Use Cases

The following use cases (which include TQL query examples) describe how the adapter can be used:

- A user needs to display all unplanned changes to all hosts running a specific application during the last 24 hours:



- ▶ A user needs to see all open critical incidents on an application and its hosts:



Viewing the Actual State

UCMDB exposes a Web Service for the use of Service Manager. The Web Service receives the CMDB ID and customer ID as an input and returns extended data for the CI, which includes properties and related CIs.

The call to the Web Service is done in the Actual State tab in HP Service Manager, when Service Manager is configured to work with UCMDB.

The Web Service executes the query in the **Integration\SM Query** folder that matches the type of CI sent. If more than one matching query exists, an exception is thrown.

The layout that is defined in the TQL query is the layout that is synchronized.

It is common for some parts of the executed query to be federated (for example, from DDMi, Asset Manager, SMS, and so on).

This section also includes:

- ▶ "Predefined Queries" on page 623
- ▶ "Configuration" on page 623

Predefined Queries

Out-of-the-box queries are located in the **Integration\SM Query** folder. Queries are selected according to the class type of the CI.

- ▶ **hostExtendedData** – used for retrieving real time extended information (Asset, Person, WindowsService, Printer, InstalledSoftware, and CPU) about a certain CI of type Node.
- ▶ **applicationExtendedData** – used for retrieving real time extended information about Business Applications.
- ▶ **businessServiceExtendedData** – used for retrieving real time extended information about Business Services.

Configuration

WSDL and XML Schema URLs for the Web Service

WSDL:

```
http://[machine_name]:8080/axis2/services/ucmdbSMService
```

XML Schema:

```
http://[machine_name]:8080/axis2/services/ucmdbSMService?xsd=xsd0
```

Manipulating the Result Using Transformations

In some cases you may want to apply additional transformations to the resulting XML (for example, to sum up all the disks' sizes and add those as an additional attribute to the CI). To add invoke additional transformation on the TQL results, place a resource named **[tql_name].xslt** in the adapter configuration as follows: **Adapter Management > ServiceDeskAdapter7-1 > Configuration Files > [tql_name].xslt**.

There is a resource named **example_calculated_attribute.xslt** that demonstrates how to sum the disk sizes using xslt.

Using Global IDs

It is possible to use the Global ID instead of the CMDB ID to work with the Actual State flow. This may be needed in multiple CMDB environments, where a none-CMS UCMDB is integrated with Service Manager. To use global IDs instead of CMDB IDs, navigate to **Adapter Management > ServiceManagerAdapterX-X > Configuration Files > sm.properties** and set **use.global.id=true**.

For details about multiple CMDB environments, see "Integrating Multiple CMDBs" in the *HP Universal CMDB Data Flow Management Guide*

If CIs were previously pushed to Service Manager from a different CMDB instance, duplicates may occur, as the CIs will not reconcile.

Compressing Location Topology to an Attribute

Due to the limitation of the Data Push flow, it is not possible to push topologies that have CIs that are not connected directly to the Root. To be able to push locations to Service Manager, an enrichment is used to concatenate the location topology to a single attribute (Calculated Location) on the Node.

The enrichments are found in the **Location** folder:

- ▶ Location_1Enrichment
- ▶ Location_2Enrichment
- ▶ Location_3Enrichment

The xslt transformer then inflates the attribute back to three different XML tags with the following xslt code:

```
<xsl:variable name="calculatedLocation" select="@calculated_location"/>
  <Building>
    <xsl:value-of select="substring-after($calculatedLocation,' Building:')"/>
  </Building>
  <Floor>
    <xsl:value-of
select="substring-before(substring-after($calculatedLocation,'Floor:'),' Building:')"/>
  </Floor>
  <Room>
    <xsl:value-of
select="substring-before(substring-after($calculatedLocation,'Room:'),' Floor:')"/>
  </Room>
```

The serviceDeskConfiguration.xml File

The `serviceDeskConfiguration.xml` Adapter configuration file contains three parts:

- 1** The first part, which is defined by the `ucmdbClassConfigurations` element, contains the external CIT configuration that the Adapter supports. For details, see "External CITs Configuration" on page 626.
- 2** The second part, defined by the `reconciliationClassConfigurations` element, contains reconciliation data information for appropriate UCMDB CITs. For details, see "Reconciliation Data Configuration" on page 628.
- 3** The third part, defined by the `globalConnectorConfig` element, includes the global configuration for a specific connector implementation. For details, see "Global Configuration" on page 632.

This section also includes the following topics:

- "External CITs Configuration" on page 626
- "Reconciliation Data Configuration" on page 628
- "Global Configuration" on page 632

External CITs Configuration

Each CIT that is supported by the adapter is defined in the first section of the adapter configuration file.

This section, `ucmdbClassConfiguration`, represents the only supported CIT configuration. This element contains the CIT name as defined in the UCMDDB class model (the `ucmdbClassName` attribute), mapping for all its attributes (the `attributeMappings` element), and a private configuration for a specific connector implementation (the `classConnectorConfiguration` element):

- The `ucmdbClassName` attribute defines the UCMDDB class model name.
- The `attributeMappings` element contains `attributeMapping` elements.

The `attributeMapping` element defines the mapping between the UCMDDB model attribute name (the `ucmdbAttributeName` attribute) to an appropriate ServiceCenter/Service Manager attribute name (the `serviceDeskAttributeName` attribute).

For example:

```
<attributeMapping ucmdbAttributeName="problem_brief_description"
serviceDeskAttributeName="brief.description"/>
```

This element can optionally contain the following converter attributes:

- The `converterClassName` attribute. This is the converter class name that converts the UCMDDB attribute value to the ServiceDesk attribute value.
- The `reversedConverterClassName` attribute. This is the converter class name that converts the ServiceDesk attribute value to the UCMDDB attribute value.

- The `classConnectorConfiguration` element contains the configuration for the specific connector implementation for the current external CIT. Wrap this configuration in CDATA if it contains special XML characters (for example, `&`; replacing `&`).

The useful fields of the Service Manager `classConnectorConfiguration` element are as follows:

- The `device_key_property_names` element contains the fields names in the WSDL information of the current object that can contain the device ID (for example, `ConfigurationItem`). Each field should be added as a `device_key_property_name` element.
- The `id_property_name` element contains the field name in the WSDL information that contains the ID of the current object.

The following example shows the `ucmdbClassConfiguration` section of the `serviceDeskConfiguration.xml` file. The section includes the `ucmdbClassName` element for the Incident CIT with a ServiceCenter connector implementation:

```
<ucmdbClassConfiguration ucmdbClassName="it_incident">
  <attributeMappings>
    <attributeMapping ucmdbAttributeName="incident_id"
serviceDeskAttributeName="IncidentID"/>
    <attributeMapping ucmdbAttributeName="incident_brief_description"
serviceDeskAttributeName="BriefDescription"/>
    <attributeMapping ucmdbAttributeName="incident_category"
serviceDeskAttributeName="Category"/>
    <attributeMapping ucmdbAttributeName="incident_severity"
serviceDeskAttributeName="severity"/>
    <attributeMapping ucmdbAttributeName="incident_open_time"
serviceDeskAttributeName="OpenTime"/>
    <attributeMapping ucmdbAttributeName="incident_update_time"
serviceDeskAttributeName="UpdatedTime"/>
    <attributeMapping ucmdbAttributeName="incident_close_time"
serviceDeskAttributeName="ClosedTime"/>
    <attributeMapping ucmdbAttributeName="incident_status"
serviceDeskAttributeName="IMTicketStatus"/>
  </attributeMappings>
</ucmdbClassConfiguration>
```

```

<classConnectorConfiguration>
  <![CDATA[ <class_configuration
connector_class_name="com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.servi
ceCenterConnector.impl.SimpleServiceCenterObjectConnector">
  <device_key_property_names>
  <device_key_property_name>ConfigurationItem</device_key_property_name>
</device_key_property_names>
<id_property_name>IncidentID</id_property_name>
<keys_action_info>
  <request_name>RetrieveUcmdbIncidentKeysListRequest</request_name>
<response_name>RetrieveUcmdbIncidentKeysListResponse</response_name>
</keys_action_info>
<properties_action_info>
  <request_name>RetrieveUcmdbIncidentListRequest</request_name>
  <response_name>RetrieveUcmdbIncidentListResponse</response_name>
</properties_action_info>
</class_configuration> ]]>
</classConnectorConfiguration>
</ucmdbClassConfiguration>

```

Adding an Attribute to a CIT

To add an attribute to the UCMDB model for an adapter-supported CIT:

- 1 Navigate to **Data Flow Management > Adapter Management >** and select the **ServiceManagerAdapter** that corresponds to your version of Service Manager.
- 2 Select **Configuration Files > ServiceDeskConfiguration.xml** file and add an **attributeMapping** element to the appropriate **ucmdbClassConfiguration** element.
- 3 Verify that ServiceCenter/Service Manager externalizes this attribute in its Web Service API.
- 4 Click **Save**.

Reconciliation Data Configuration

Each UCMDB CIT that can be related to the adapter-supported CIT is defined in the second section of the **serviceDeskConfiguration.xml** file.

This section, `reconciliationClassConfigurations`, represents the reconciliation data configuration for one UCMDB CIT. The element includes two attributes:

- The `ucmdbClassName` attribute. This is the CIT name as defined in the UCMDB class model.
- The `concreteMappingImplementationClass` attribute. This is the class name of the concrete implementation for the `ConcreteMappingEngine` interface. Use this attribute to map between instances of UCMDB CITs and external Adapter CITs. The default implementation that is used is:

```
com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.mapping.impl.OneNodeMappingEngine
```

An additional implementation exists that is used only for the host reconciliation CIT for reconciliation by the IP of the host:

```
com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.mapping.impl.HostIpMappingEngine
```

The `reconciliationClassConfiguration` element can contain one of the following elements:

- The `reconciliationById` element. This element is used when the reconciliation is done by ID. In this case, the text value of this element is the ServiceDesk field name that contains the CMDB ID. For example:

```
<reconciliationById>UcmdbID</reconciliationById>
```

In this example, the ServiceDesk field `UcmdbID` contains the CMDB ID of the appropriate host.

- The `reconciliationData` element. Use this element if the reconciliation is done by comparing attributes. You can run reconciliation with one attribute or several attributes by using the logical operators OR and/or AND.

If you run reconciliation with one attribute, the `reconciliationData` child element should be a `reconciliationAttribute` element. The `reconciliationAttribute` element contains an appropriate UCMDB attribute name (the `ucmdbAttributeName` attribute) and an appropriate ServiceDesk attribute name (the `serviceDeskAttributeName` attribute). This element can also contain a `ucmdbClassName` attribute that defines the appropriate UCMDB CIT name. By default, the current reconciliation UCMDB CIT name is used.

You can also use the `converterClassName` and `reversedConverterClassName` attributes; they should contain the converter class name that converts the UCMDB attribute value to the ServiceDesk attribute value, or vice versa.

For example:

```
<reconciliationData>
  <reconciliationAttribute ucmdbAttributeName="name"
    serviceDeskAttributeName="NetworkName"
    converterClassName="com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.con
    verter.PropertyValueConverterToUpperCase"/>
</reconciliationData>
```

For reconciliation to run with two or more attributes, use a logical operator between reconciliation attributes.

The logical operator AND can contain several `reconciliationAttribute` elements (the minimum is 2). In this case the reconciliation rule contains an AND operator between attribute comparisons.

For example:

```
<reconciliationData>
<AND>
  <reconciliationAttribute ucmdbAttributeName="name"
    serviceDeskAttributeName="NetworkName"
    converterClassName="com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.con
    verter.PropertyValueConverterToUpperCase"/>
  <reconciliationAttribute ucmdbClassName="ip_address"
    ucmdbAttributeName="name" serviceDeskAttributeName="NetworkAddress" />
</AND>
</reconciliationData>
```

In this example, the reconciliation rule follows this format: `node.name= NetworkName` and `ip_address.name= NetworkAddress`.

The logical operator OR can contain several `reconciliationAttribute` and AND elements. In this case, the reconciliation rule contains an OR operator between attributes and AND expressions. Since XML does not assure the order of elements, you should provide a priority attribute to each sub-element of OR element type. The comparison between OR expressions is calculated by these priorities.

For example:

```
<reconciliationData>
<OR>
    <reconciliationAttribute ucmdbAttributeName="primary_dns_name"
serviceDeskAttributeName="NetworkDNSName" priority="2" />
<AND priority="1" >
    <reconciliationAttribute ucmdbAttributeName="name"
serviceDeskAttributeName="NetworkName"
converterClassName="com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.con
verter.PropertyValueConverterToUpperCase"/>
    <reconciliationAttribute ucmdbClassName="ip_address"
ucmdbAttributeName="name" serviceDeskAttributeName="NetworkAddress" />
</AND>
</OR>
</reconciliationData>
```

In this example the reconciliation rule follows this format: `(node.primary_dns_name= NetworkDNSName OR (node.name= NetworkName and ip_address.name= NetworkAddress))`. Since the AND element takes a priority attribute of value 1, the `(node.name= NetworkName and ip_address.name= NetworkAddress)` condition is checked first. If the condition is satisfied, the reconciliation is run. If not, the `.host_dnsname= NetworkDNSName` condition is checked.

The additional sub-element of the `reconciliationClassConfiguration` element is `classConnectorConfiguration`. The `classConnectorConfiguration` element contains the configuration for a specific connector implementation for the current reconciliation CIT. This configuration should be wrapped by CDATA if it contains some special XML characters (for example, `&`; replacing `&`).

Changing the Reconciliation Rule of a CIT

- 1 In `serviceDeskConfiguration.xml`, update the appropriate `reconciliationData` element with the new rule.
- 2 Call to the JMX to reload the adapter: **FCmdb Config Services > loadOrReloadCodeBaseForAdapterId**, using the appropriate customer ID and `ServiceDeskAdapter` adapter ID, or go to the Integration Points pane and reload the adapter from there. For details, see "Integration Point Pane" in the *HP Universal CMDB Data Flow Management Guide*.

Reconciliation of a Host by ip_address or by name

To run reconciliation on a host by **ip_address** or **name**, place the following `ReconciliationData` element in the Adapter configuration file:

```
<reconciliationData>
  <OR>
    <reconciliationAttribute priority="1" ucmdbClassName="ip_address"
ucmdbAttributeName="ip_address" serviceDeskAttributeName="NetworkAddress"/>
    <reconciliationAttribute priority="2" ucmdbClassName="node"
ucmdbAttributeName="name" serviceDeskAttributeName="NetworkName"
converterClassName="com.mercury.topaz.fcmdb.adapters.serviceDeskAdapter.converter.PropertyValueConverterToUpperCase"/>
  </OR>
</reconciliationData>
```

Global Configuration

The third section of the Adapter configuration file contains the global configuration for the specific connector implementation. This configuration, `globalConnectorConfig`, should be wrapped by `CDATA` if it contains some special XML characters (for example, `&` replacing `&`).

The useful fields of the Service Manager `globalConnectorConfig` element are as follows:

- 1 The `date_pattern` element contains the date adapter with which the Service Manager works.

The default is `MM/dd/yy HH:mm:ss`.

If the date adapter is wrong, an FTQL returns wrong date condition results.

- 2 The **time_zone** element defines the time zone of Service Manager. The default is the UCMDB server time zone.

To check the Service Manager date adapter and time zone:

- a **Service Manager version 7:** Access **Menu Navigation > System Administration > Base System Configuration > Miscellaneous > System Information Record**. Click the **Date Info** tab.
 - b **ServiceCenter version 6.1:** Access **Menu Navigation > Utilities > Administration > Information > System Information**. Click the **Date Info** tab.
- 3 The **max_query_length** element defines the maximal query length in a Service Manager Web service request. The default value is 1000000.
 - 4 The **name_space_uri** element defines the name space URI to connect to the Service Manager Web service. The default value is `http://servicecenter.peregrine.com/PWS`.
 - 5 The **web_service_suffix** element defines the Service Manager Web service center URI suffix. The default value is `sc62server/ws`. It is used when the URL is created.

Tasks

Deploy the Adapter – Typical Deployment

This section describes a typical deployment of the adapter.

This task includes the following steps:

- 1** "Deploy the ServiceDesk Adapter" on page 634
 - a** "Add a ServiceCenter/Service Manager External Data Source" on page 635
 - b** "Configure HP ServiceCenter 6.2" on page 637 (when connecting to HP ServiceCenter)
 - c** "Configure HP Service Manager 7.0/7.1" on page 639 (when connecting to HP Service Manager)
- 2** "Add an Attribute to the ServiceCenter/Service Manager CIT" on page 640
 - a** "Add an Attribute to the UCMDB Model" on page 651
 - b** "Export Attributes from HP ServiceCenter by Changing the Configuration" on page 641 (when connecting to HP ServiceCenter)
 - c** "Export Attributes from HP Service Manager by Changing the Configuration" on page 643 (when connecting to HP Service Manager)
 - d** "Modify the Adapter Configuration File" on page 646

Deploy the ServiceDesk Adapter

This section explains where to place the files needed for deployment.

This task includes the following steps:

- "Add a ServiceCenter/Service Manager External Data Source" on page 635
- "Configure HP ServiceCenter 6.2" on page 637
- "Configure HP Service Manager 7.0/7.1" on page 639

1 Add a ServiceCenter/Service Manager External Data Source

In this step, you add an integration point.

- a** In UCMDB, select **Data Flow Management > Integration Studio**.
- b** Click the **Create New Integration Point** button to add an integration point. Select the **ServiceDeskAdapter** that matches your version of Service Manager and fill in the mandatory fields.



For help with this dialog box, see "Create New Integration Point/Edit Integration Point Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

- c** To select the appropriate attributes for the CI Type, click the Federation tab. For details, see "Federation Tab" in the *HP Universal CMDB Data Flow Management Guide*.
- d** In the Create New Integration Point dialog box, enter the following information:

Enter the following information:

Name	Recommended Value	Description
Adapter	<user defined>	Select Service Center 6.2x, Service Manager 7.0x, or Service Manager 7.1x as required.
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> .
Hostname/IP	<user defined>	The name of the server on which HP Service Manager is running
Integration Name	Service Desk Target Adapter	The name you give to the integration point.

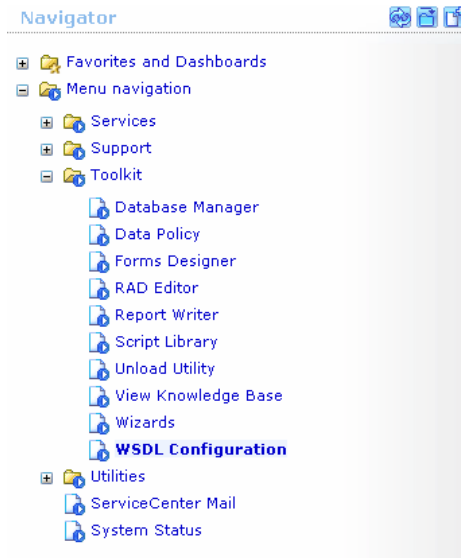
Name	Recommended Value	Description
Is Integration Activated	selected	Select this checkbox to create an active integration point. You clear the checkbox if you want to deactivate an integration, for instance, to set up an integration point without actually connecting to a remote machine.
Port	<user defined>	The server port at which HP Service Manager is connected.

- e** Click **Test connection**, verify the status of the Connection Test Status, and click **OK**.
- f** Click **Next** and verify that the following message is displayed: **A connection has been successfully created.**
- g** Continue with "Configure HP ServiceCenter 6.2" on page 637 or "Configure HP Service Manager 7.0/7.1" on page 639.

2 Configure HP ServiceCenter 6.2

If you are connecting to HP ServiceCenter 6.2, perform the following procedure. If you are connecting to HP Service Manager 7.0/7.1, skip this step.

- a Open HP ServiceCenter, then the ServiceCenter client.
- b Display **WSDL Configuration** in the Navigator (**Main Menu > Menu navigation > Toolkit**):



- c In the Name field, enter **device** and press **Enter**:

Search External Access Definition Records

Back Add Search Find Fill

External Access Definition

Service Name:

Name: Object Name:

Allowed Actions Expressions Data Policy

Allowed Actions	Action Names
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

- d Select the **Data Policy** tab and ensure that the `network.name` attribute is not empty (its value should be **NetworkName**). Change the value to **false**. Save your changes.

Service Name:

Name:

Object Name:

Allowed Actions Expressions **Data Policy**

Field Name	API Caption	Exclude	API Data Type
mac.address		true	
manufacturer		true	
model	Model	false	
mtbf		true	
network.address		true	
network.name	NetworkName	false	
nm.id		true	
nondevice		true	
objid		true	
operating.system		true	
order.line.item		true	

- e After saving, click the **Cancel** button.
- f In the Object Name field type **Change** and press **Enter**.
- g Select the Data Policy tab and ensure that:
 - The **header,coordinator** attribute is not empty (its value should be **Coordinator**). Change the value to **false**.

Service Name:

Name:

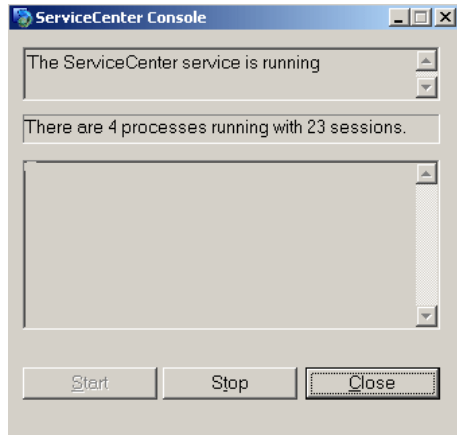
Object Name:

Allowed Actions Expressions **Data Policy**

Field Name	API Caption	Exclude	API Data Type
header,company	Company	false	
header,coord.date		true	
header,coord.dept		true	
header,coord.phone	CoordinatorPhone	false	
header,coordinator	Coordinator	false	

- The **header,orig.operator** attribute is not empty (its value should be **OpenedBy**). Change the value to **false**.
- h Save the changes.

- i Restart ServiceCenter: Select **Start > Programs > ServiceCenter 6.2 > Server > Console** to open the ServiceCenter Console.



- j Click **Stop** and then **Start**.
- k Continue with "Add an Attribute to the UCMDB Model" on page 651.

3 Configure HP Service Manager 7.0/7.1

If you are connecting to HP Service Manager 7.0/7.1, perform the following procedure. If you are connecting to HP ServiceCenter 6.2, skip this step.

- a Import the unload file relevant to the Service Manager version with which you are working: **ucmdbIntegration7_0x.unl** or **ucmdbIntegration7_1x.unl**. To do so, in Service Manager, click **Menu Navigation > Tailoring > Database Manager**.
 - Right-click the detail button and select **Import/Load**.
 - In the HP Service Manager File Load/Import page, click **Specify File** and browse to the following unload file:

C:\hp\UCMDBServer\runtime\fcmdb\CodeBase\ServiceManager Adapter7-1

The file is loaded via the file browser.
 - Enter the description in the **Import Description** box.

- ▶ Select **winnt** in the **File Type** list.
 - ▶ Select a display option.
 - ▶ Click **Load FG** to start loading.
- b** Continue with "Add an Attribute to the UCMDB Model" on page 651.

Add an Attribute to the ServiceCenter/Service Manager CIT

This section explains how to retrieve additional data from ServiceCenter or Service Manager by adding an attribute to the CIT.

This task includes the following steps:

- ▶ "Add an Attribute to the UCMDB Model" on page 651
- ▶ "Export Attributes from HP ServiceCenter by Changing the Configuration" on page 641
- ▶ "Export Attributes from HP Service Manager by Changing the Configuration" on page 643
- ▶ "Modify the Adapter Configuration File" on page 646

1 Add an Attribute to the UCMDB Model

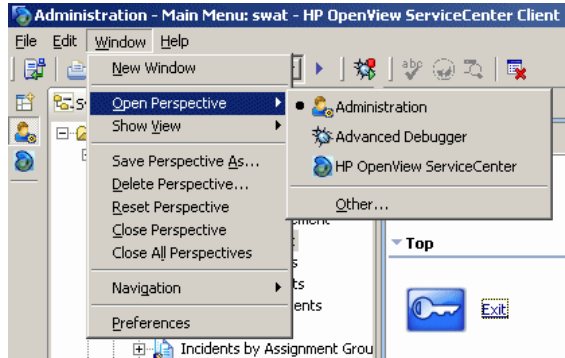
Edit the Incident CIT to add the new attribute to UCMDB as follows:

- a** Navigate to **Modeling > CI Type Manager**.
- b** In the CI Types pane, select **IT Process Record > Incident**.
- c** Select the Attributes tab and add the new attribute.
- d** Continue with "Export Attributes from HP ServiceCenter by Changing the Configuration" on page 641 or "Export Attributes from HP Service Manager by Changing the Configuration" on page 643.

2 Export Attributes from HP ServiceCenter by Changing the Configuration

If you are connecting to HP ServiceCenter, perform the following procedure.

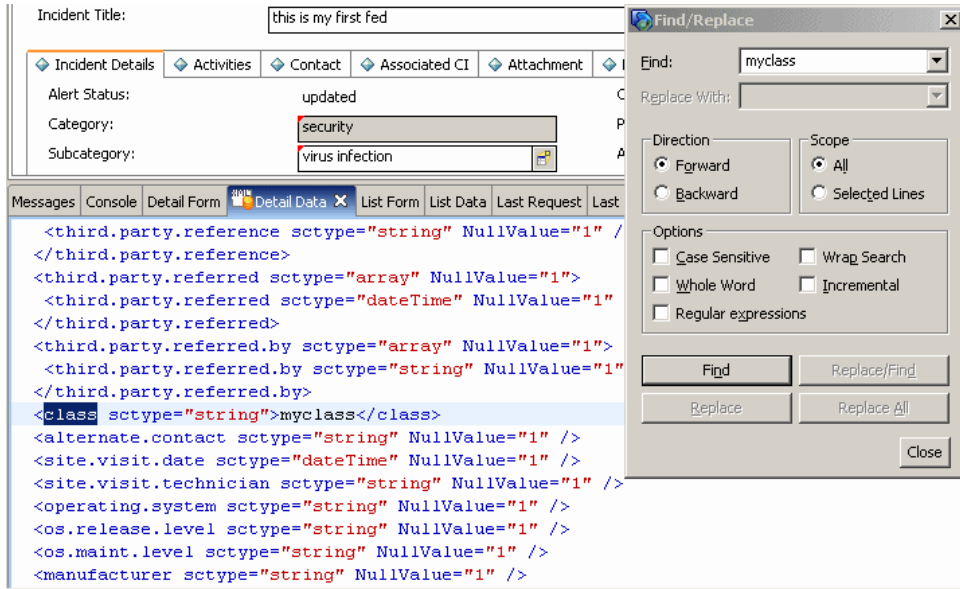
- a In HP ServiceCenter, open the ServiceCenter client.
- b Select **Window > Open Perspective > Administration:**



- c Select **Incident Management > All Open Incidents**, and select one of the incidents you created.

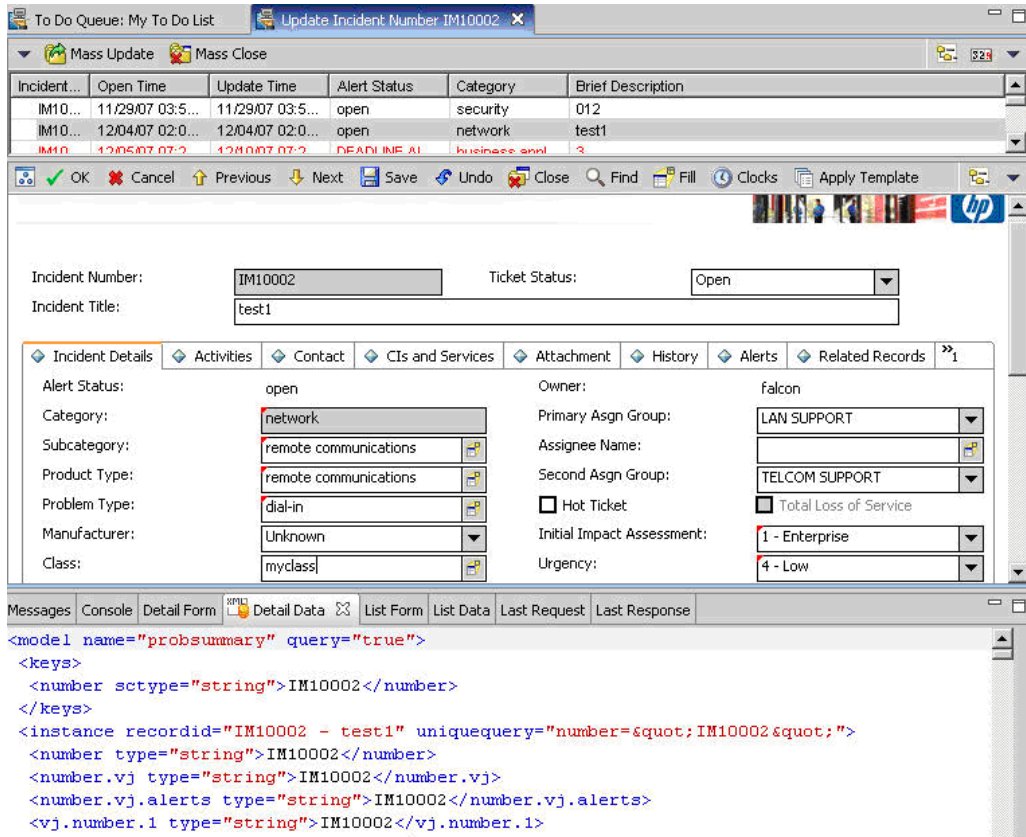
Note: Verify that the value in the Class field is the one that you want to report to UCMDB.

- d Search for the value you entered in the Class field (that is, **myclass**), in the XML file displayed below. This is the CI name in ServiceCenter.



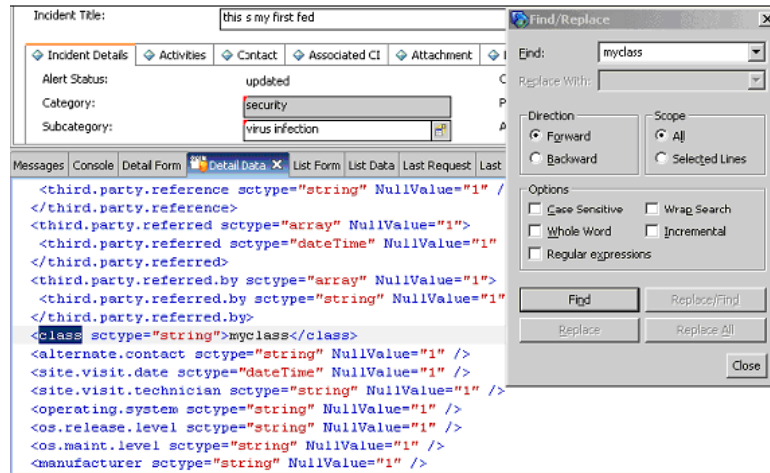
- e Display **WSDL Configuration** in the Navigator (**Main Menu > Menu navigation > Toolkit**). Locate the Object Name field, enter **Incident** and press **Enter**.
- f Select the **Data Policy** tab. Enter a name for the CI mentioned in the XML file (that is, **class**). Change the value to **false**. Save your changes.
- g Restart ServiceCenter: Select **Start > Programs > ServiceCenter 6.2 > Server > Console** to open the ServiceCenter Console.
- h Click **Stop** and then **Start**.
- i Continue with "Modify the Adapter Configuration File" on page 646.

- b** Open one of the incidents you created: Select **Incident Management > Search Incidents**. Click the search button (you can filter the fields to limit the search).



Note: Verify that the value in the Class field is the one that you want to report to HP Universal CMDB.

- c Search for the value you entered in the Class field (that is, **myclass**), in the XML file displayed below. This is the CI name in Service Manager.



- d Display **WSDL Configuration** in the Navigator (**Main Menu > Menu Navigation > Tailoring**). Locate the Object Name field, enter **UcmdbIncident** and press **Enter**.
- e Select the **Data Policy** tab.
- f Select the **Fields** tab and ensure that the CI name mentioned in the XML file (that is, **class**) appears in the Field list with **ClassName** as its caption. If this attribute does not appear in the Field list, add it and save your changes.
- g Continue with "Modify the Adapter Configuration File" on page 646.

4 Modify the Adapter Configuration File

Perform this procedure for all configurations.

- a** Navigate to **Data Flow Management > Adapter Management** and select the **ServiceManagerAdapter** that corresponds to your version of Service Manager. Continue and select **Configuration Files > ServiceDeskConfiguration.xml**.
- b** Edit the **ServiceDeskConfiguration.xml** file by navigating to **Data Flow Management > Adapter Management > ServiceManagerAdapter** (the one that corresponds to your version of Service Manager) > **Configuration Files > ServiceDeskConfiguration.xml**
- c** Add the new attribute line under the Incident area: Locate the following marker:

```
<ucmdbClassConfiguration ucmdbClassName="it_incident">  
<attributeMappings>
```

- d** Add the following line:

```
<attributeMapping ucmdbAttributeName="incident_class"  
ServiceDeskAttributeName="ClassName"/>
```

where:

- **ucmdbAttributeName="incident_class"** is the value defined in the CI Type Manager
 - **ServiceDeskAttributeName="ClassName"** is the valued defined in ServiceCenter/Service Manager
- e** Click **Save**.

Communicate with Service Manager over SSL

The following procedure explains how to open communication with Service Manager over SSL.

This task includes the following steps:

- ▶ "Add an SM Self-signed Certificate to the UCMDB Trusted Stores" on page 647
- ▶ "Add the SM External Data Source Using Communication Over SSL" on page 648

1 Add an SM Self-signed Certificate to the UCMDB Trusted Stores

- a** Copy the SM self-signed certificate to a directory. (To export SM self-signed certificates, refer to the Service Manager documentation).
- b** Locate the JRE security folder, by default located in:
C:\hp\UCMDB\UCMDBServer\bin\jre\lib
- c** Back up the **cacerts** file by renaming it.
- d** Open a command line window and execute the following commands (to import the previously created or copied certificate):

For HP Universal CMDB 8.0x:

```
cd C:\hp\UCMDB\UCMDBServer\jre\bin"
keytool.exe -import -keystore
C:\hp\UCMDB\UCMDBServer\j2f\JRE\lib\security\cacerts" -trustcacerts -file
<full path to SM self-signed certificate>
```

For HP Universal CMDB 9.00 or later:

```
cd C:\hp\UCMDB\UCMDBServer\bin\jre\bin
keytool.exe -import -keystore
C:\hp\UCMDB\UCMDBServer\bin\jre\lib\security\cacerts -trustcacerts -file
<full path to SM self-signed certificate>
```

- e** Restart the UCMDB service.

2 Add the SM External Data Source Using Communication Over SSL

- a In UCMDB, navigate to **Data Flow Management > Integration Studio**.
- b Define an integration point using the following parameters: In the Create New Integration Point dialog box, choose the **ServiceDeskAdapter** for your version of ServiceCenter or Service Manager, and enter the user name, password, and URL. The URL field should contain: **https://<SM server name>:13443/sc62server/ws**.

For details, see "Create New Integration Point/Edit Integration Point Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

Set Up Service Manager Integration for Data Push

This task describes how to define an integration using the ServiceCenter/Service Manager adapter to push data to an external data repository.

This task includes the following steps:

- "Define an Integration Point" on page 648
- "Define a Data Push Changes Job" on page 649
- "Define a Data Push RMI job" on page 650
- "Run the Jobs" on page 650

1 Define an Integration Point

For details, see "Add a ServiceCenter/Service Manager External Data Source" on page 635.

2 Define a Data Push Changes Job

Use any of the following out-of-the-box queries (located in the **Integration\SM Sync** folder) to create a job of type Changes (for pushing CI queries):

- **hostData** – use to push nodes. Pushed data includes nodes whose NodeRole attribute is either empty, 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). Due to limitations of the Changes flow, the location information is saved using an enrichment in the Calculated Location attribute.
- **networkData** – use to push nodes that are not pushed with the **hostData** query. This query is similar to **hostData**, except that it pushes nodes whose NodeRole attribute is not empty and does not contain the following strings: desktop, server, virtualized_system, or printer.
- **printerData** – use to push printers (network printers). This query is similar to **networkData**, except that it does push nodes where the NodeRole attribute contains the string printer.
- **applicationData** – use to push Business Applications.
- **businessServiceData** – used to push Business Services.

For details, see "Integration Jobs Pane" in the *HP Universal CMDB Data Flow Management Guide*.

All CI attributes that are pushed should have Change Monitored set (STATIC qualifier) in order to be written to the History so that changes are caught. Each relation must have the qualifier TRACK_LINK_CHANGES in order to be written to the history. Link and attribute changes that are not written to history are not detected as changed.

Note:

- ▶ Select the Allow Delete check box if you want your Data Push job to send deletes of CIs & Links to Service Manager.
 - ▶ The Changes flow is required for integration with Service Manager because it creates a single CI out of a topology, which matches the Service Manager specification.
-

3 Define a Data Push RMI job

Use any of the following out-of-the-box queries (located in the **Integration\SM Sync** folder) to create a job of type RMI (for pushing Relation queries):

- ▶ **hostRelationsData** – use to push Layer2 (Physical) connections between pairs of nodes through their interfaces.
- ▶ **applicationRelationsData** – use to push logical relations between Business Applications to other Business Applications and nodes.
- ▶ **businessServiceRelationsData** – use to push logical relations between Business Services to other Business Services, applications and nodes.

For details, see "Integration Jobs Pane" in the *HP Universal CMDB Data Flow Management Guide*.

4 Run the Jobs

- a Run the Changes Job, and then run the RMI job.
- b Click the **Statistics** button (**Data Flow Management > Integration Studio**) to review the jobs' statistics. Compare the statistics to the TQLs by using the **Calculate Query Result Count** button in the Modeling Studio.
- c In Service Manager, verify that the CIs have been pushed correctly.



Add a New Attribute to an Existing CI Type

Perform the following steps to add a new attribute to an existing CI type.

This task includes the following steps:

- "Add an Attribute to the UCMDB Model" on page 651
- "Add the Attribute to the Layout of the TQL Query" on page 651
- "Map the Attribute in the SM Adapter Configuration" on page 651
- "Map the Field in the Service Manager Web Service" on page 652

1 Add an Attribute to the UCMDB Model

- a** Navigate to **Modeling > CI Type Manager**.
- b** Select the CI type to which you want to add the attribute.
- c** Select the Attributes tab and add the new attribute.

2 Add the Attribute to the Layout of the TQL Query

- a** Navigate to **Modeling > Modeling Studio**.
- b** Select the query that contains the CI type you want to change (located in the **Integration\SM Sync** folder).
- c** Right-click the node of the CI type you are changing and select **Query Node Properties**.

3 Map the Attribute in the SM Adapter Configuration

- a** Navigate to **Data Flow Management > Adapter Management** and select the ServiceManagerAdapter that corresponds to your version of Service Manager.
- b** Select Configuration Files, and choose the xslt file that contains the CI type you changed.
- c** Add the attribute at the file.device XML tag or at the concrete file XML tag of the type (depends on the Service ManagerWeb Service).

4 Map the Field in the Service Manager Web Service

For details, refer to the Service Manager documentation.

Add a New CI Type

Perform the following steps to add a new CI type to the UCMDB class model.

This task includes the following steps:

- ▶ "Add the CI Type to the UCMDB Class Model" on page 652
- ▶ "Define a TQL Query for Synchronizing the CI Type" on page 652
- ▶ "Map the Attribute in the SM Adapter Configuration" on page 653
- ▶ "Map the CI Type in the SM Adapter Configuration" on page 653
- ▶ "Create and Map the Field in the Service Manager Web Service" on page 653
- ▶ "Update the Data Push Job" on page 653

1 Add the CI Type to the UCMDB Class Model

- a Navigate to **Modeling > CI Type Manager**.
- b Add the new CI type and its valid relations.

2 Define a TQL Query for Synchronizing the CI Type

- a Navigate to **Modeling > Modeling Studio**.
- b In the **Integration\SM Sync** folder, create a new query.

The new TQL query should include the new CI type (which should be labeled as **Root**) and all the related CIs that are connected to the root node for the additional data. For example: in the **hostData** TQL query, **IpAddress** and **Interface** are the additional data of the node.

The TQL query should also contain the layout that you want to synchronize.

3 Map the Attribute in the SM Adapter Configuration

- a** Navigate to **Data Flow Management > Adapter Management** and select the ServiceManagerAdapter that corresponds to your version of Service Manager.
- b** Select Configuration Files, and choose the xslt file that contains the CI type you changed.
- c** Add the attribute at the file.device XML tag or at the concrete file XML tag of the type (depends on the Service Manager Web Service).

4 Map the CI Type in the SM Adapter Configuration

- a** Navigate to **Data Flow Management > Adapter Management** and select the ServiceManagerAdapter that corresponds to your version of Service Manager.
- b** Select Configuration Files.
- c** Create a new xslt file for the new CI type and map all the attributes and related CIs to it.
- d** Open **smSyncConfFile.xml** and add a mapping between the new TQL query and the new xslt file.

5 Create and Map the Field in the Service Manager Web Service

For details, refer to the Service Manager documentation.

6 Update the Data Push Job

- a** Navigate to **Data Flow Management > Integration Studio**.
- b** Configure the Data Push job you created to include the new TQL query.

Reference

Flow and Configuration

The ServiceCenter/Service Manager adapter receives data and a TQL definition from the Data Push engine, transforms it into a SOAP call for each instance of the TQL query's results, and sends the SOAP requests to Service Manager.

The transformation between the UCMDB class model to the Service Manager class model is done by an XSLT engine.

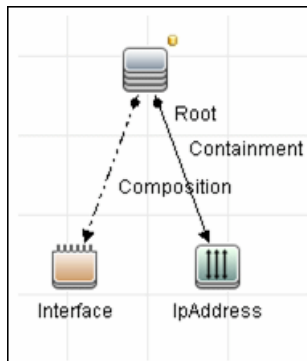
This section also includes:

- ▶ "Parse the TQL Definition" on page 654
- ▶ "XSLT transformation" on page 658

Parse the TQL Definition

The TQL definition must have one Root node (in which case it will be considered a CI synchronization TQL) or several Root links (in which case it will be considered a Relations synchronization TQL).

Example of an out-of-the-box TQL query for synchronizing a node CI type:



To XML

The result of the TQL query is divided into instances according to the Root node/links, and each instance is given an XML representation.

XML Schema

Each TQL query is automatically assigned a schema according to the structure of the TQL adapter and the layout attributes chosen.

Example of an XML schema for a TQL query:

This example displays the XML schema for a TQL query using a UCMDB JMX located at [http://\[cmdb_machine\]:8080/jmx-console/HtmlAdaptor, service=FCmdb Config Services, createXMLSchemaFromTql\(](http://[cmdb_machine]:8080/jmx-console/HtmlAdaptor,service=FCmdb%20Config%20Services,createXMLSchemaFromTql()

```
java.lang.String createXMLSchemaFromTql()
```

Produce XML Schema for a tql. XML with this schema will be used in some target adapters for transformation purposes.

Param	ParamType	ParamValue	ParamDescription
customerId	int	1	Customer id
tqlName	java.lang.String	applicationData	Name of the TQL Query

Invoke

XML schema for a networkData TQL query example:

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="node">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ip_addressss" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ip_address" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="friendlyType" type="xs:string"/>
                  <xs:attribute name="id" type="xs:string"/>
                  <xs:attribute name="ip_netmask" type="xs:string"/>
                  <xs:attribute name="name" type="xs:string"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```



```

        <xs:element name="interfaces" minOccurs="0" maxOccurs="1">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="interface" minOccurs="0"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:attribute name="friendlyType" type="xs:string"/>
                            <xs:attribute name="id" type="xs:string"/>
                            <xs:attribute name="mac_address" type="xs:string"/
>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="calculated_location" type="xs:string"/>
            <xs:attribute name="default_gateway_ip_address" type="xs:string"/>
            <xs:attribute name="discovered_os_name" type="xs:string"/>
            <xs:attribute name="discovered_os_version" type="xs:string"/>
            <xs:attribute name="friendlyType" type="xs:string"/>
            <xs:attribute name="global_id" type="xs:string"/>
            <xs:attribute name="id" type="xs:string"/>
            <xs:attribute name="node_role" type="xs:string"/>
            <xs:attribute name="primary_dns_name" type="xs:string"/>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

Example of XML for a networkData TQL query:

```

<node customer_id="1" discovered_os_name="windows 2010"
  discovered_os_version="build45-2a" friendlyType="Net Device"
  global_id="bdef388c1b1b3db863ce442a96b54e53"
id="bdef388c1b1b3db863ce442a96b54e53"
  default_gateway_ip_address="1.2.3.4"
  calculated_location="Room:234 Floor:2 Building:M54"
node_role="&lt;Values&gt;&lt;Value&gt;firewall&lt;/Value&gt;&lt;/Values&gt;"
primary_dns_name="myDNS.com">
  <ip_addresss direction="outgoing" linkType="Containment">
    <ip_address customer_id="1" friendlyType="IpAddress"
      id="91757d9d45f166437c1864e931f59e16" ip_address="16.59.64.1"/>
    <ip_address customer_id="1" friendlyType="IpAddress"
      id="f91bf4c40b06e460b51af2178181843d" ip_address="16.59.66.1"/>
  </ip_addresss>
</node>

```

XSLT transformation**Mapping a TQL name to XSLT**

To map between the TQL names and the XSL files, navigate to **Data Flow Management > Adapter Management > ServiceManagerAdapter** (the one that corresponds to your version of Service Manager) > **Configuration Files > smSyncConfFile.xml**.

Example of XML for configuring a hostData TQL query:

The file includes the names of the Service Manager requests for each operation (create, update, and delete).

```

<tql name="hostData" xslFile="host_data.xslt">
  <!-- this is host->ip,interface,sm_host tql -->
  <request type="Create" name="CreateucmdbComputerRequest"/>
  <request type="Update" name="UpdateucmdbComputerRequest"/>
  <request type="Delete" name="DeleteucmdbComputerRequest"/>
</tql>

```

The **smSyncConfFile.xml** file must be updated when you add a new TQL query that will be synchronized with Service Manager.

Result after transformation

This sample shows the results after **host_data1.xslt** is executed on the original XML file.

```

<UpdateucmdbNetworkRequest>
  <model>
    <keys/>
    <instance>
      <file.device>
        <UCMDBId>bdef388c1b1b3db863ce442a96b54e53</UCMDBId>
        <CustomerId>1</CustomerId>
        <Subtype>firewall</Subtype>
        <Building>M54</Building>
        <Floor>2</Floor>
        <Room>234</Room>
        <DefaultGateway>1.2.3.4</DefaultGateway>
        <OS>windows 2010</OS>
        <DNSName>myDNS.com</DNSName>
      </file.device>
      <file.networkcomponents>
        <OSVersion>build45-2a</OSVersion>
        <addIIPAddr>
          <addIIPAddr>
            <AddIIPAddress>16.59.64.1</AddIIPAddress>
            <AddSubnet/>
          </addIIPAddr>
          <addIIPAddr>
            <AddIIPAddress>16.59.66.1</AddIIPAddress>
            <AddSubnet/>
          </addIIPAddr>
        </addIIPAddr>
      </file.networkcomponents>
    </instance>
  </model>
</UpdateucmdbNetworkRequest>

```

XSLT references

XSLT is a standard language for transforming XML documents into other XML documents. The adapter uses the built-in Java 1.5 Xalan XSLT 1.0 transformer. For details about XSLT see:

<http://www.w3.org/TR/1999/REC-xslt-19991116>

<http://www.w3schools.com/xsl/>

<http://www.zvon.org/xxl/XSLTutorial/Output/index.html>

Reuse of XSLT parts

In addition to the standard XSLT specifications, the adapter? supports the use of an XSLT preprocessor that scans XSL files for comments such as `<!--import:[file_name]-->` in the XSLT, and replaces them with the contents of `[file_name]`.

Service Manager WSDL

Tools such as SoapUI or SoapSonar can be used to view the WSDL files.

Service Manager Web Services are dynamic and can be modified. For details on how to edit or add new Service Manager Web Services, refer to the Service Manager documentation.

Service Manager Result SOAP request

For details on how to enable printing of SOAP requests, see "Logs" on page 662.

Using Mapping Tools

An automatic tool (such as Mapforce) can be used to create XSLT mappings between the CMDB XML schema and the Service Manager XML schema.

Troubleshooting and Limitations

This section describes troubleshooting and limitations for the ServiceCenter/Service Manager adapter.

Changes Flow Limitations

- ▶ A query should contain one CI that is labeled as Root or one or more relations that are labeled as Root_<postfix>.

The root node is the main CI that is synchronized, and the other nodes are the contained CIs of the main CI. For example, when synchronizing Nodes, the query node of (Node) will be labeled as Root and the host resources will not be root.

- ▶ The TQL graph must not contain cycles.
- ▶ The TQL query must only contain the Root CI, and optionally CIs that are directly connected to it.
- ▶ A query that is used to synchronize relations should have cardinality 1...* and OR condition between them.
- ▶ Any conditions must reside on the Root CI only.
- ▶ If you want to synchronize only specific Roots from a TQL query, you must configure the required condition on these Roots, and then, configure the same condition in the TQL that synchronize the relationships that are linked to the Roots.
- ▶ Compound relations are not supported.
- ▶ Subgraphs are not supported.
- ▶ if one of the TQL queries that are used for synchronization (including layout changes) is edited, the changes will not be synchronized until a full data push job has been manually run. Results from a previous synchronization will not be deleted from the Service Manager server.
- ▶ Changes to NodeRole only will not be detected and will not update CI for the next Data Push job.

Logs

Use the **fcmdb.adapters.log** file to troubleshoot the Service Desk adapter (located in the **UCMDBServer\runtime\log** folder).

To view the complete SOAP request and response in addition to other information, use the **fcmdb.properties** file to change the adapter's log level to debug: **log4j.category.fcmbd.adapters=debug,fcmbd.adapters**.

Do not forget to change the log level back to **error** when you are finished debugging.

For example, if the **fcmbd.adapters.log** of an Service Manager integration names SM01, for each single CI sent the log will show:

```
DEBUG - SM01 >> Source CI tree is: (The XML as outputted by the ucmbd goes here)
INFO - SM01 >> ===== start run soap message
INFO - SM01 >> ===== create urs required time = 0
DEBUG - SM01 >> Run message: (The XML Send after Xslt Transformation goes here)
DEBUG - SM01 >> Response message: (The XML response goes here)
INFO - SM01 >> ===== stop run soap message. The required time = 390
```

In multi-threaded push flows the thread name indicates the chunk number and thread number:

```
[SM01_pushObjectWorkerThread-<ChunkID>::<ThreadID>]
```

Actual State

To troubleshoot the Actual State flow, use a SOAP testing tool such as SoapUI or SoapSonar to run a SOAP request similar to this:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://
www.w3.org/2001/XMLSchema" xmlns:types="http://schemas.hp.com/ucmbd/1/types">
  <soap:Body>
    <types:getAllCIProperties>
      <types:ID>17868889fd660853e16a474e10df5de3</types:ID>
    </types:getAllCIProperties>
  </soap:Body>
</soap:Envelope>
```

You will obtain a response similar to this:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header />
  <soapenv:Body>
    <types:getAllCIPPropertiesResponse xmlns:types="http://schemas.hp.com/ucmdb/1/
types">
      <types:CI id="17868889fd660853e16a474e10df5de3" type="Windows">
        <types:prop type="string">
          <types:name>Host Name</types:name>
          <types:value>LABM2AM209</types:value>
        </types:prop>
        <types:prop type="string">
          <types:name>Host Operating System</types:name>
          <types:value>Windows 2003 Server Enterprise Edition </types:value>
        </types:prop>
        <types:prop type="string">
          <types:name>Host Vendor</types:name>
          <types:value>Microsoft Windows</types:value>
        </types:prop>
        <types:prop type="string">
          <types:name>Host DNS Name</types:name>
          <types:value>labm2am209.devlab.ad</types:value>
        </types:prop>
        <types:prop type="string">
          <types:name>Asset Tag</types:name>
          <types:value>GB8718DS72__</types:value>
        </types:prop>
        <types:complexProp className="IP" size="1">
          <types:item>
            <types:prop type="string">
              <types:name>IP Address</types:name>
              <types:value>16.59.56.161</types:value>
            </types:prop>
            <types:prop type="string">
              <types:name>IP Network Mask</types:name>
              <types:value />
            </types:prop>
          </types:item>
        </types:complexProp>
      ...
    </types:getAllCIPPropertiesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</types:CI>  
  </types:getAllCIPropertiesResponse>  
</soapenv:Body>  
</soapenv:Envelope>
```

If errors occur, review the following files for exceptions:

- **C:\hp\UCMDB\UCMDBServer\runtime\log\error.log**
- **C:\hp\UCMDB\UCMDBServer\runtime\log\cmdb.operation.log**

42

Network Node Manager *i* (NNMi) Integration

This chapter includes:

Concepts

- ▶ NNMi Integration Overview on page 666
- ▶ NNMi - UCMDB Integration Architecture on page 668

Tasks

- ▶ Set Up HP NNMi–HP UCMDB Integration on page 669
- ▶ Run HP NNMi–UCMDB Integration on page 670
- ▶ Use the HP NNMi–HP UCMDB Integration on page 678
- ▶ Change the HP NNMi–HP UCMDB Integration Configuration on page 681
- ▶ Disable HP NNMi–HP UCMDB Integration Configuration on page 681
- ▶ Perform Impact Analysis on page 683

Reference

- ▶ HP NNMi–HP UCMDB Integration Configuration Form Reference on page 684

Troubleshooting and Limitations on page 688

Concepts

NNMi Integration Overview

You integrate NNMi with UCMDB using the Data Flow Management (DFM) application.

When you activate the **Discovery-Based Product Integrations > NNM Layer 2** module, DFM retrieves Layer 2 network topology data from NNMi and saves the data to the UCMDB database. Users can then perform change management and impact analysis.

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 666
- ▶ "Supported Versions" on page 667

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:

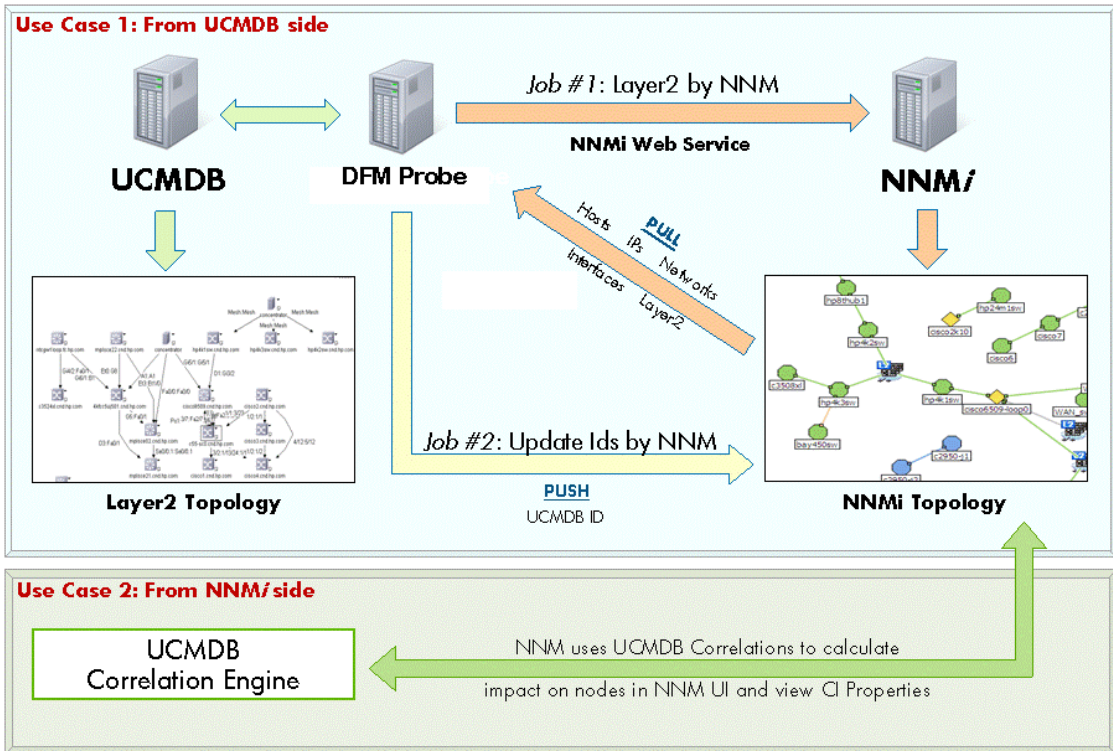
- Data Flow Probe version 9.00 or later
- HP NNMi version 8.11 or later

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 8.00 through 8.x.

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 669
- "Customize the Integration" on page 670

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 685.
- 4** Enter the information for connecting to the UCMDB server. For information about these fields, see "UCMDB Server Connection" on page 686.
- 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 687.
- 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 671
- "Set Up the NNMi Protocol" on page 673
- "Activate the Discovery Jobs" on page 674
- "Check Messages for Successful Job Execution" on page 676
- "Discovered CITs" on page 676
- "Topology Map" on page 677
- "Validation of Results" on page 677

1 Prerequisites

- Ensure that the Data Flow Probe is installed, as detailed in the *HP Universal CMDB Deployment Guide* PDF.
- NNMi integration jobs are triggered against the **IpAddress** CI of the NNMi server. This **IpAddress** CI must be present in UCMDB. This **IpAddress** CI may be discovered in one of the following ways:
 - "Discover the IpAddress CI of the NNMi Server" on page 671
 - "Manually Add the IpAddress CI of the NNMi Server" on page 672

After the **IpAddress** CI has been discovered, perform the step "Verify CI Discovery" on page 673.

Note: When you installed HP Universal CMDB or 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 **IpAddress** CI automatically. Therefore, you should create this CI manually in the CMDB, as described in "Manually Add the IpAddress CI of the NNMi Server."

Discover the IpAddress CI of the NNMi Server

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. For details, see "Add/Edit IP Range Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.

To discover the IP CI of the NNMi server:

- 1 Navigate to **Data Flow Management > Discovery Control Panel**.
- 2 In the **Network Discovery - Basic** module, select the **Range IPs by ICMP** job and click the **Properties** tab. Locate the **Parameters** pane.
- 3 In the **Range** parameter line, select the **Override** check box and add the IP address of the NNMi server. Click **OK** to save the job.

- 4 Activate the job to discover the **IpAddress** CI of the NNMi server.

Manually Add the IpAddress CI of the NNMi Server

Note: When you installed HP Universal CMDB or 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 **IpAddress** CI. If any other license (Basic or Advanced) is deployed on the UCMDB server, use the "Discover the IpAddress CI of the NNMi Server" procedure.

To manually add the IpAddress CI of the NNMi server

- 1 Verify that the Data Flow Probe is correctly installed and connected to the UCMDB Server.
- 2 Add the IP of the NNMi server to the Data Flow 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. For details, see "Add/Edit IP Range Dialog Box" in *HP Universal CMDB Data Flow Management Guide*.
- 3 Insert the **Address** CI of the NNMi server in the CMDB:
 - a Navigate to **Modeling > IT Universe Manager**.
 - b In the CI Selector pane, click the **Browse Views** tab and select **Network Topology** from the **View** drop-down menu.
 - c Click the **New CI** button.



- d** In the **New CI** dialog box, select the **Address** CIT from the tree and enter the following values:

Field	Description
IP Address	The IP address of the NNMi server.
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 **Address** CI.

Verify CI Discovery

Note: Verification of CI discovery is relevant only when the **Address** CI of the NNMi server is discovered (as described in "Discover the IPAddress CI of the NNMi Server" on page 671), not when it is added manually.


In HP Universal CMDB, verify that the **Address** CI of the NNMi server (through the ICMP jobs) has been discovered before running the NNMi discovery.



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** Access **Data Flow Management > Data Flow Probe Setup**.
- b** In the Domains and Probes tree, navigate to the Domain for which you want to set up the NNMi protocol, and click **Credentials**.
- c** Select **NNM Protocol** and click .

- d** Set the protocol attributes and click **OK**. For details, see "NNM Protocol" in the *HP Universal CMDB Data Flow Management Guide*.

3 Activate the Discovery Jobs

The NNMi jobs are included in the **Discovery-Based Product Integrations > NNM Layer 2** module.

► Layer2 by NNM

This job connects to the NNMi Web service and retrieves NNMi discovered nodes, IPs, networks, interfaces, physical ports, VLANs, hardware boards, and Layer 2 connection information to create a Layer 2 topology in UCMDB.

► Update Ids in NNM

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.

Note: Because the NNMi Web service enables updating of only one node at a time, 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. Check **probeMgr-adaptersDebug.log** for the update status.

To activate the Layer2 by NNM job:

- a** Navigate to **Data Flow Management > Discovery Control Panel**.
- b** In the **Discovery-Based Product Integrations > NNM Layer2** module, select the **Layer2 by NNM** job and click the **Properties** tab.
- c** Right-click the job name and select **Activate**.
- d** In the Discovery Status pane, click the **Add CI** button.



- e** In the **Choose CIs to Add** dialog box, search for the **Address** CI of the NNMi server and click **Add**.
- f** Click **Close**. The job is activated against the selected **Address** CI of the NNMi server.

To activate the Update Ids in NNM job:

- a** Navigate to **Data Flow Management > Discovery Control Panel**.
- b** In the **Discovery-Based Product Integrations > NNM Layer 2** module, select the **Update Ids in NNM** job.
- c** Right-click the job name and select **Activate**.



- d** In the Discovery Status pane, click the **Add CI** button.
- e** In the **Choose CIs to Add** dialog box, search for the **Address** CI of the NNMi server and click **Add**.
- f** Click **Close**. The job is activated against the selected **Address** CI of the NNMi server.

4 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_Utils: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:

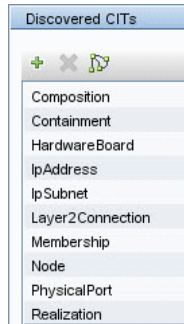
```
- The Job 'NNM Update IDs' started invocation (on 1 destinations)
- UCMDB Server: ucmdb75.fkam.cup.hp.com, UCMDB Port: 8080, UCMDB Username:
admin, UCMDB Protocol: http, UCMDB Context: /axis2/services/UcmdbService
- NNM Server: it2tst10.cnd.hp.com, NNM Port: 80, NNM Username: system
- Getting ready 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.
```

5 Discovered CITs

To view discovered CITs, select a specific adapter in the Resources pane.

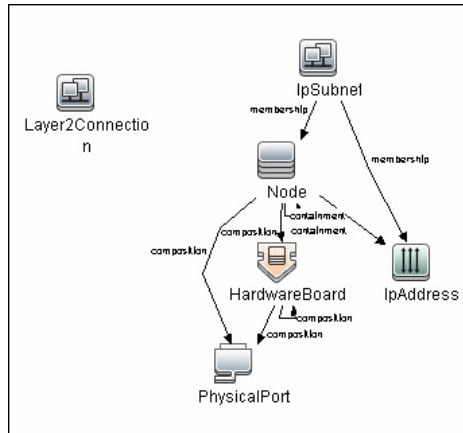
For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

The Layer2 by NNM job:



6 Topology Map

The Layer2 by NNM job:



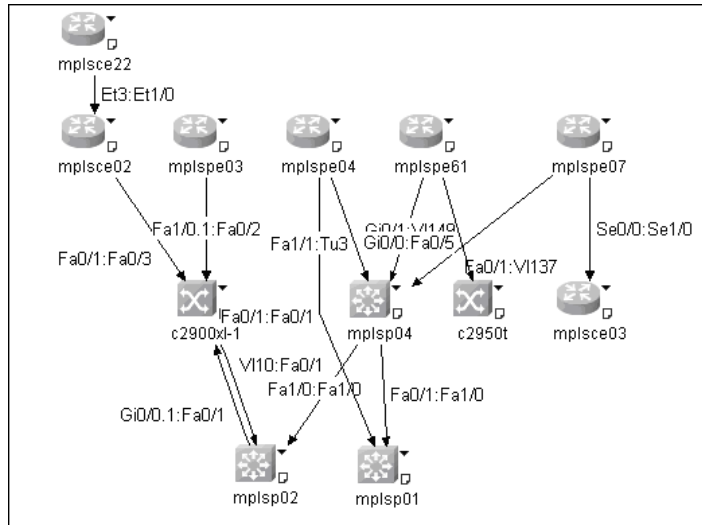
7 Validation of Results

Verify that data was discovered using the NNMi integration jobs.

a For the **Layer 2 by NNM** job:

- In UCMDB, navigate to **Admin > Modeling > IT Universe Manager**.
- In the **CI Selector** pane, select **View Browser**.

- In the **View** drop-down menu, select **Layer 2**. Select a view. The view displays the CIs and relationships discovered by the integration job:



- b** For the **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: this attribute 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 679.
- The **Open CI in UCMDB** action, which is described in "View the UCMDB CI" on page 680.

For information about using the integration from the UCMDB user interface, see "Run HP NNMi–UCMDB Integration" on page 670.

View Impacted CIs

Testing for impacted configuration items in UCMDB involves firing a test event of the designated severity and then evaluating the specified impact analysis rules to determine if the event impacts any other configuration items.

For example:

- ▶ Impact analysis 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.
- ▶ Impact analysis 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 impact analysis rules, see "Impact Analysis Manager" in the *HP Universal CMDB Modeling Guide*.

For the HP NNMi–HP UCMDB integration, the parameters described in "Integration Behavior" on page 687 specify the severity of the test event and the group of UCMDB impact analysis 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**.

General Properties	
CMDB ID:	47152a32a903776fc986e76ec379398f
CI type:	Application
Updated By:	UCMDB: User:admin
City:	
Name: *	Loan_Application_Application
Deletion Candidate Period:	20
Origin:	
<input type="checkbox"/> Is Update By Owner	
Display Label:	Loan_Application
<input checked="" type="checkbox"/> Allow CI Update	
User Label:	Loan_Application
Actual Deletion Period:	40
Country:	
Created By:	enrichment.Loan Application
Note:	Test notes... blah blah blah.
Description:	Test loan application dependent on NNM managed switch
State:	
Update Time:	7/3/08 11:57 AM
Create Time:	6/13/08 8:14 PM

Other Properties	
Application ID: *	642687

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 684.
- 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:

- 1** In the NNMi console, open the **HP NNMi–HP UCMDB Integration Configuration** form (**Integration Module Configuration > HP UCMDB**).
- 2** 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 impact analysis rules 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 impact analysis, see "Impact Analysis Manager" in the *HP Universal CMDB 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 685
- ▶ "UCMDB Server Connection" on page 686
- ▶ "Integration Behavior" on page 687

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 685
- ▶ "UCMDB Server Connection" on page 686
- ▶ "Integration Behavior" on page 687

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	<p>The user name for connecting to the NNMi console. This user must have the NNMi Administrator or Web Service Client role.</p>
HP NNMi Password	<p>The password for the specified NNMi user.</p>

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	<p>The fully-qualified domain name of the UCMDB server.</p>

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 impact analysis rules that the Find UCMDB Impacted CIs action runs to calculate impact. The default prefix of NNM_ corresponds to the default UCMDB impact analysis 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 analysis 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.

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 adapter 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 Adapter**.
- b** Click the **Adapter 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 adapter, then reactivate the job against the **IpAddress** CI of the NNMi server.

43

Storage Essentials (SE) Integration with HP Universal CMDB

This chapter includes:

Concepts

- ▶ SE Integration – Overview on page 692

Tasks

- ▶ Discover the SE Oracle Database on page 693

Reference

- ▶ Storage Essentials Integration Packages on page 695
- ▶ Discovered CITs on page 695
- ▶ Views on page 700
- ▶ Impact Analysis Rules on page 704
- ▶ Reports on page 706

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 693
- "Network and Protocols" on page 694
- "Activate the Discovery Job" on page 694

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.
-

Network and Protocols

SE uses the **SQL protocol**. For details, see "SQL Protocol" in *HP Universal CMDB Data Flow Management Guide*.

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*.

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 impact analysis 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.

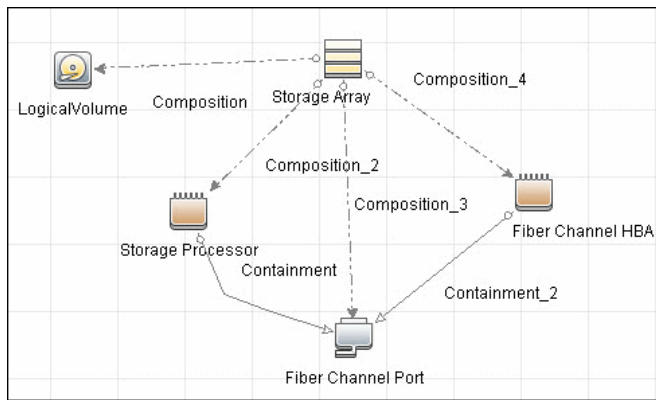
This CIT retrieves Storage Array details from the **mvc_storagesystemssummaryvw** view. DFM retrieves detailed information on Storage Processors and HBAs from the **mvc_storageprocessorssummaryvw** and **mvc_cardsummaryvw** tables respectively.

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.

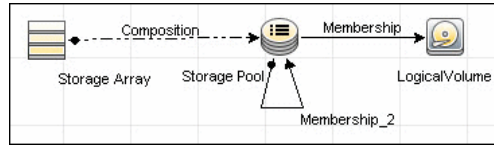
Results from these queries populate a map as shown below:



- ▶ **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:



To view discovered CITs, select a specific adapter in the Resources pane.

For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.

Host Details

DFM retrieves Host details from the **mvc_hostsummaryvw** view and detailed information on HBAs from the **mvc_cardsummaryvw** view.

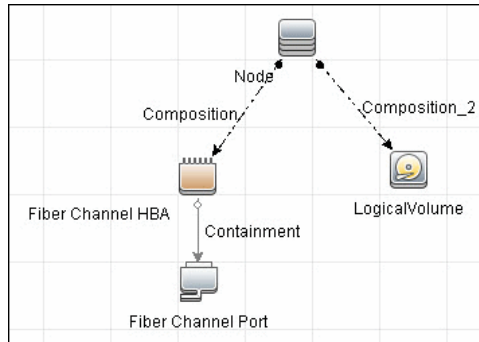
SE maintains information on Operating Systems, Memory, IP address, and DNS name on each host. DFM uses this information to create Host CIs of type UNIX or Windows, and adds Memory CIs for each host as available.

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 hosts 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.

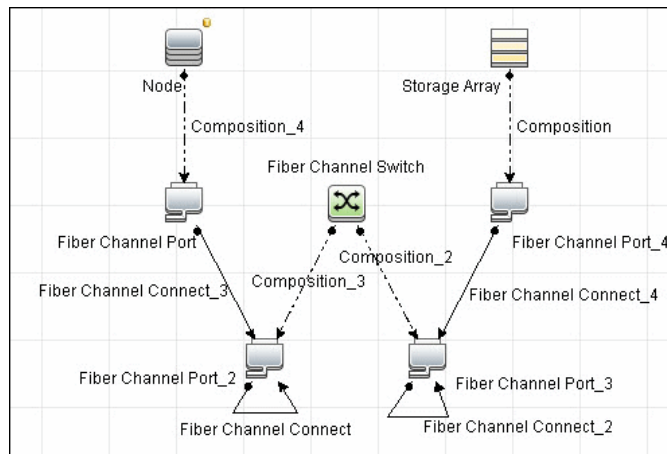
Results from these queries populate a map as shown below:



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.

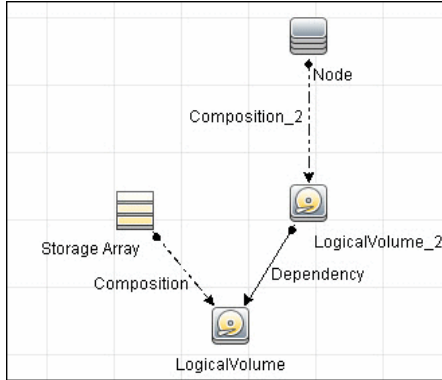
Results from these queries populate a map as shown below:



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:



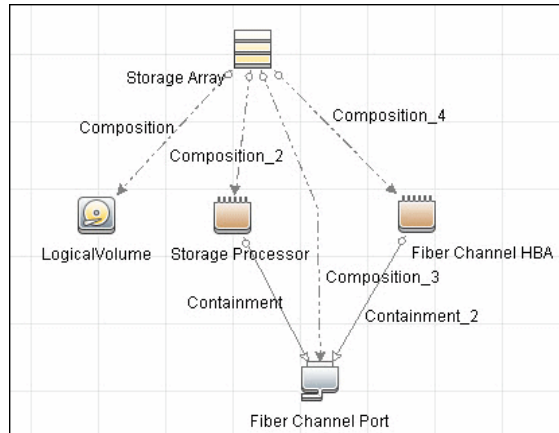
Views

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

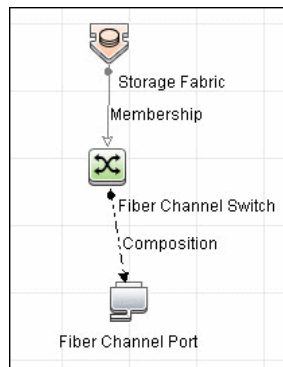
This view shows a Storage Array and its components including Logical Volumes, HBAs, Storage Processors, and Fiber Channel Ports. The view shows each component under its container Storage Array and groups Logical Volumes by CI Type.

Storage Array does not require all components in this view to be functional. Composition 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.



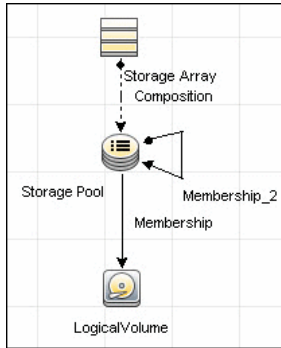
FC Switch Details

This view shows a Fiber Channel Switch and all connected Fiber Channel Ports.



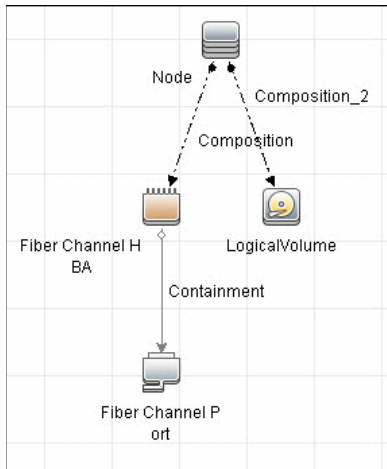
Storage Pool Details

This view shows Storage Pools with associated Storage Arrays and Logical Volumes.



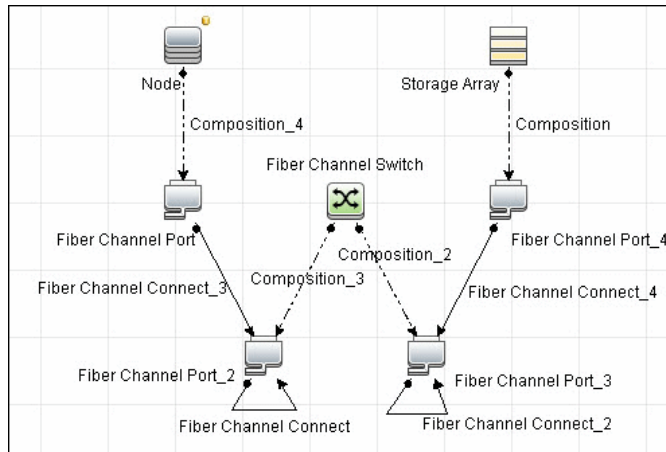
Host Storage Details

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.



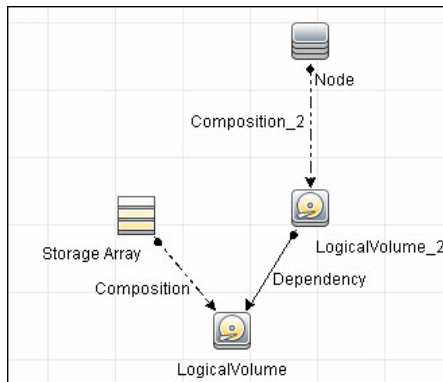
SAN Topology

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.



Impact Analysis Rules

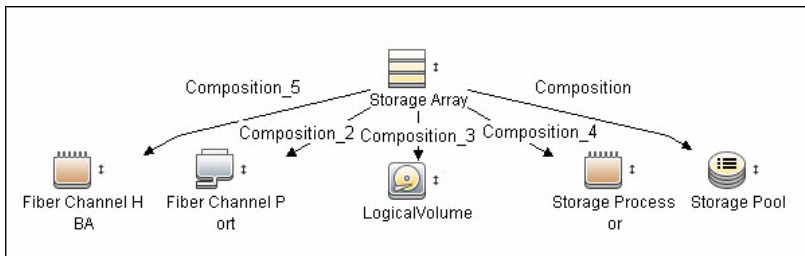
This package contains basic impact analysis rules to enable impact analysis and root cause analysis in UCMDB. These impact analysis rules are templates for more complex rules that you can define based on business needs.

All impact analysis rules fully propagate both Change and Operation events. For details on impact analysis, see "Impact Analysis Manager Page" and "Impact Analysis Manager Overview" in the *HP Universal CMDB Modeling Guide*.

Note: Impact analysis events are not propagated to Fiber Channel Ports for performance reasons.

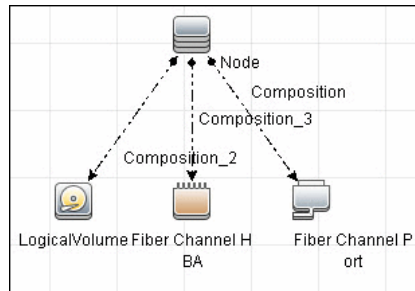
Storage Array Devices to Storage Array

This impact analysis rule propagates events between Logical Volumes, Storage Processors, Fiber Channel HBAs, and Storage Arrays.



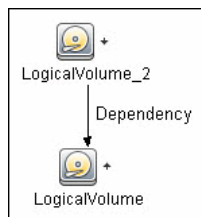
Host Devices to Host

This impact analysis rule propagates events between Fiber Channel HBAs and Hosts, and Logical Volumes on the Host.



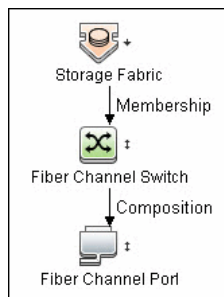
Logical Volume to Logical Volume

This impact analysis rule propagates events on a Logical Volume contained in a Storage Array to the dependent Logical Volume on the Host.



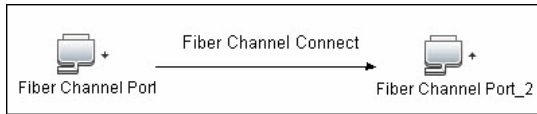
FC Switch Devices to FC Switch

This impact analysis 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 impact analysis 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 impact analysis 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.

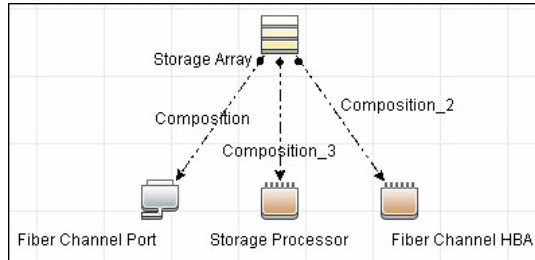
Reports

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 707, "Host Configuration" on page 707, "Storage Array Dependency" on page 708, and "Host Storage Dependency" on page 708.

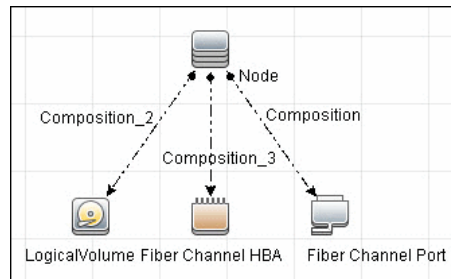
Storage Array Configuration

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.



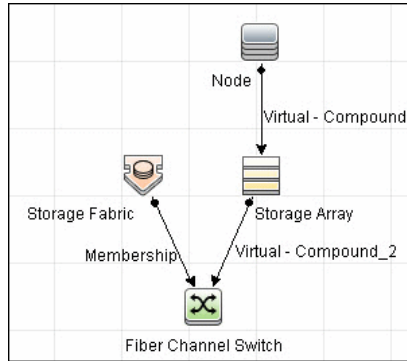
Host Configuration

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.



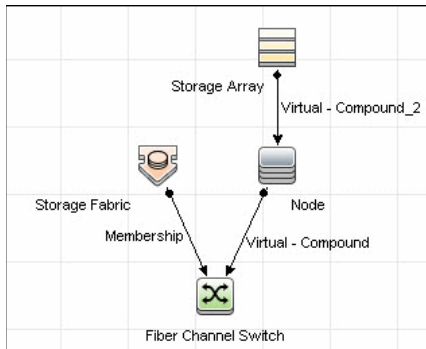
Storage Array Dependency

This report maps dependencies on a Storage Array. The report also displays information on switches connected to it.



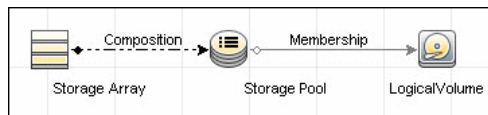
Host Storage Dependency

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.



44

HP Systems Insight Manager (HP SIM) Integration

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 710
- ▶ Discovery Mechanism on page 710

Tasks

- ▶ Discover HP SIM Data Center Infrastructure on page 714

Reference

- ▶ Instance Views on page 723

Troubleshooting and Limitations on page 725

Concepts

Overview

HP Universal CMDB (UCMDB) can discover data center infrastructure information stored in an HP Systems Insight Manager (HP SIM) system. Integration involves synchronizing devices, topology, and the hierarchy of a data center infrastructure in the UCMDB database (CMDB). This enables change management and impact analysis across all business services mapped in UCMDB, from an infrastructure point of view.

UCMDB initiates discovery on the HP SIM server through Web service calls. Synchronized configuration items (CIs) include nodes such as Windows, and UNIX servers, network devices, printers, clusters, cellular/partitioned systems, blade enclosures, and racks. Some server components, for example, CPU and memory, are also synchronized. The integration also synchronizes relationships between blade servers and blade enclosures, virtual machines, physical servers, and so on. The synchronization uses an XML-based mapping that dynamically changes synchronized CIs and attributes without requiring a code change.

For details on nodes and attributes in HP SIM, refer to the Database tables section of the *HP SIM Technical Reference* guide.

Discovery Mechanism

Data Flow Management (DFM) uses the HP SIM Web service API to retrieve node information from the HP SIM database. DFM also enables you to specify extended attributes that should be retrieved for each node.

To enable inclusion in a UCMDB spiral discovery schedule, discovery is split into two jobs. The **SIM WebService Ports** job triggers on all IpAddress CIs in the CMDB and looks for port 50001—the port at which HP SIM listens for Web service queries. The **SIM Integration by WebService** job triggers on results from the **SIM WebService Ports** job and retrieves data.

HP SIM represents hosts (blade enclosures, racks, servers, and so on) as Nodes; UCMDB has separate CITs for each such host. To represent hosts correctly in UCMDB, a two-level mapping is used, to enable integration customization without code changes. This makes the integration completely customizable and dynamic.

For details on jobs, see "Discovery Control Panel – Advanced Mode Workflow" in *HP Universal CMDB Data Flow Management Guide*.

This section describes the two levels of mapping:

- "Node to Host CI Type Mapping" on page 711
- "Node Attribute to CI Type and CI Attribute Mapping" on page 713

Node to Host CI Type Mapping

All IP-enabled systems are represented as **Nodes** in HP SIM and each node has attributes (for example, operating device type and operating system name) that can be used to classify nodes/hosts as specific CITs in UCMDB. The first level of mapping involves setting parameters on the **SIM Integration by WebServices** job. This job includes **HostCitIdentifierAttributes** and **HostCitIdentifierMap** parameters that are used for the mapping:

- **HostCitIdentifierAttributes**. This attribute specifies the names of HP SIM Node attributes that are used for the mapping. This parameter uses the **DeviceType** and **OSName** out-of-the-box Node attributes. The parameter accepts comma-separated node attribute names, is case sensitive, and expects each node attribute name to be enclosed in single quotes.
- **HostCitIdentifierMap**. This attribute specifies the mapping between values of the above HP SIM Node attributes and corresponding UCMDB CITs. This parameter accepts a comma-separated list of value pairs, where each value pair takes the following format:

```
'node attribute value':'UCMDB CI Type'
```

Both attributes are case-sensitive and must be enclosed in single quotes. Each Node-attribute value is one possible value of one or more Node attribute names specified in the **HostCitIdentifierAttributes** parameter. Each UCMDB CIT is the name (not the display name) of the UCMDB CIT to which this value maps.

This parameter has out-of-the-box mappings as follows:

HP SIM Node Attribute	UCMDB CIT
'AIX'	'unix'
'Complex'	'complex'
'Embedded'	'management_processor'
'Enclosure'	'enclosure'
'HPUX'	'unix'
'Hypervisor'	'unix'
'LINUX'	'unix'
'MgmtProc'	'management_processor'
'Printer'	'netprinter'
'Rack'	'rack'
'Server'	'node'
'Solaris'	'unix'
'Switch'	'switch'
'WINNT'	'nt'
'Workstation'	'node'

Example mapping based on the above settings:

- ▶ If the **DeviceType** attribute of a node has the value **Switch**, in UCMDB the node is represented as a **Switch** CIT.
- ▶ If the **OSName** attribute of a node has the value **WINNT**, in UCMDB the node is represented as an **NT** CIT (Display name: **Windows**).

The DFM script parses these mapping parameters from left to right and does not stop on success, so the rightmost match is considered final. This means that if a node has **DeviceName = Server** and **OSName = HPUX**, the rightmost match is **OSName** with value **HPUX**. The resulting CIT for this node in UCMDB is **unix** because **HPUX** maps to **unix**.

Node Attribute to CI Type and CI Attribute Mapping

Once the nodes are mapped to CITs using DFM job parameters as described in "Node to Host CI Type Mapping" on page 711, individual node attributes (including extended node attributes) are mapped to corresponding attributes (or CITs, as appropriate) using a generic UCMDB integration framework. The framework uses an XML file in which source and target CIT and attribute names are specified.

A sample XML mapping file (**SIM_To_UCMDB_Sample_MappingFile.xml**) that includes all host CITs mapped in the Node to Host CI Type Mapping section is included in the **SIM_Integration** package. The sample file includes host resources (for example, Memory, CPU, Disk) and relationship mapping information, to build relationships between various nodes (for example, Blade Enclosure to server, virtual machine host to guest, and so on).

Using this framework, you can map additional CITs without any code changes. For example, to map HBAs, add a new section to the XML file. Define the node attributes that identify an HBA and its attributes. Relationships between HBAs and HOSTs are also required.

Tasks

Discover HP SIM Data Center Infrastructure

This task describes how to discover data center infrastructure information stored in an HP Systems Insight Manager (HP SIM) system.

This task includes the following steps:

- "Supported Versions" on page 714
- "Prerequisites" on page 714
- "Deploy the HP SIM Package" on page 716
- "Perform Setup on the Probe Machine" on page 717
- "Set up Protocols" on page 717
- "Enable Chunking (Optional)" on page 718
- "Trigger Query for the SIM Webservice Ports Job" on page 719
- "Input Query for the SIM Webservice Ports Job" on page 719
- "Trigger Query for the SIM Integration by WebServices Job" on page 720
- "Discovery Workflow" on page 720
- "Discovered CITs – The SIM Integration by WebServices Job" on page 721
- "Discovered CITs – The SIM Webservice Ports Job" on page 722

1 Supported Versions

This discovery solution supports HP SIM versions 5.1, 5.2, 5.3, 6.0, and 6.1.

2 Prerequisites

This discovery solution includes a protocol for HP SIM.

To use the HP SIM protocols, configure the appropriate credentials, port, and trust store information to the HP SIM Web Service API. For details, see "Set up Protocols" on page 717.

Important: If you set up an HTTPS connection to connect to the SIM WebService API (that is, not an HTTP connection), the **SIM Integration by WebService** job performs no validation of any certificates presented by the HP SIM server. The job trusts any certificate issued by the HP SIM server and uses it for SSL enabled communication.

The following additional requirements must be satisfied for the mapping file to be valid for HP SIM (for details on the mapping files, see "Discovery Mechanism" on page 710):

- Verify that source and target are **HP SIM** and **HP UCMDB** respectively.
- Verify that attribute names specified in the **HostCitIdentifierAttributes** parameter are included as attributes of each host CIT in the XML file.

That is, the **OSName** and **DeviceType** attributes must be included for each **host_node** (Computer), **chassis** (Chassis), **netprinter** (Net Printer), **switch** (Switch), **nt** (Windows), **unix** (Unix), **hp_complex** (Complex), and **management_processor** (Management Processor) CIT.

- Verify that default attributes (that is, non-extended attributes) of a node have a **Node.** prefix in the mapping file.

That is, you should specify attributes such as **OSName**, **DeviceType**, and **IPAddress** as **Node.OSName**, **Node.DeviceType**, and **Node.IPAddress**.

- Verify that each Node CIT has the following attribute mapping to enable the generation of the **host_key** attribute:

```
<target_attribute name="host_key" datatype="StrProp" >
  <map type="direct" source_attribute="host_key" />
</target_attribute>
```

Note: The **host_key** attribute is the primary key attribute on **Host** and derived CITs. Since HP SIM uses a different type of key attribute, the XML definition for the **host_key** attribute is included in the mapping file, to enable generation of the **host_key** primary key attribute.

- Verify that the IP Address mapping section has the following attribute to enable automatic population of the IP domain attribute:

```
<target_attribute name="ip_domain" datatype="StrProp">  
  <map type="direct" source_attribute="ip_domain" />  
</target_attribute>
```

Note: For details on the list of HP SIM nodes and attributes, refer to the HP SIM documentation.

3 Deploy the HP SIM Package

The **SIM_Integration.zip** package contains CIT definitions, views, and trigger queries for DFM, discovery scripts, discovery adapters, and discovery jobs.

For details on deploying the package, see "Package Manager" in the *HP Universal CMDB Administration Guide*.

Note: The **SIM Integration by WebServices** job requires a JAR library to connect to the HP SIM CMS Web Service API. The versions of JAX used in this JAR conflict with those used by the UCMDDB Systinet-based Web service DFM jobs. Therefore, if Web service DFM jobs are being used, you should deploy this package on a separate Probe machine.

4 Perform Setup on the Probe Machine

- a Copy `mxpartnerlib.jar` from this directory:
`C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\hpsim`
 to this directory:
`C:\hp\UCMDB\DataFlowProbe\content\lib`
- b Open `C:\hp\UCMDB\DataFlowProbe\bin\WrapperEnv.conf` for editing.
- c Comment out line ~50 with a hash sign (#) at the beginning, for example:

```
#set.SYSTINET_CLASSES=...
```

- d Save and close the file.
- e Restart the Probe.

5 Set up Protocols

Set up the HP SIM Protocol credentials (**Data Flow Management > Data Flow Probe Setup > Domains and Probes > <domain name> Credentials > HP SIM Protocol**).

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

Note: By default, the following fields are required: **Port Number**, **SIM WebService Protocol**, **User Name**, and **User Password**. The **SIM Database ...** fields are required if the **dbIP** parameter on the discovery job is populated. For details, see "Enable Chunking (Optional)" on page 718.

6 Enable Chunking (Optional)

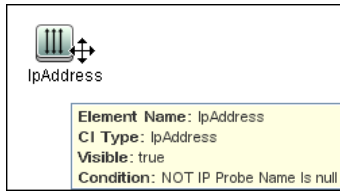
If the HP SIM server being discovered contains or manages a large number of nodes (more than 1,000), you should consider enabling chunking (**Data Flow Management > Adapter Management > select an adapter > Adapter Management tab > Adapter Parameters pane**):

Name	Value
Chunk Size	500
DebugMode	false
HostCitIdentifierAttributes	'DeviceType', 'OSName'
HostCitIdentifierMap	'Server':'host_node', 'Workstation':'host_node', 'Rack':'rac...
dbIP	

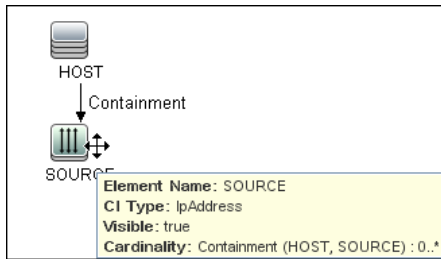
- a** To reduce load on the SIM server, if necessary, you can set the **ChunkSize** parameter in the **SIM Integration by WebServices** job to a lower value than the default **500**.
- b** Populate the **dbIP** parameter in the **SIM Integration by WebServices** job with the IP address of the HP SIM CMS database.
- c** Populate the **SIM Database ...** fields in the HP SIM protocol with connection details for the HP SIM CMS database.

Note: HP SIM CMS database details (except for the password) are located in the **Systems Insight Manager\config\database.props** file on the HP SIM server.

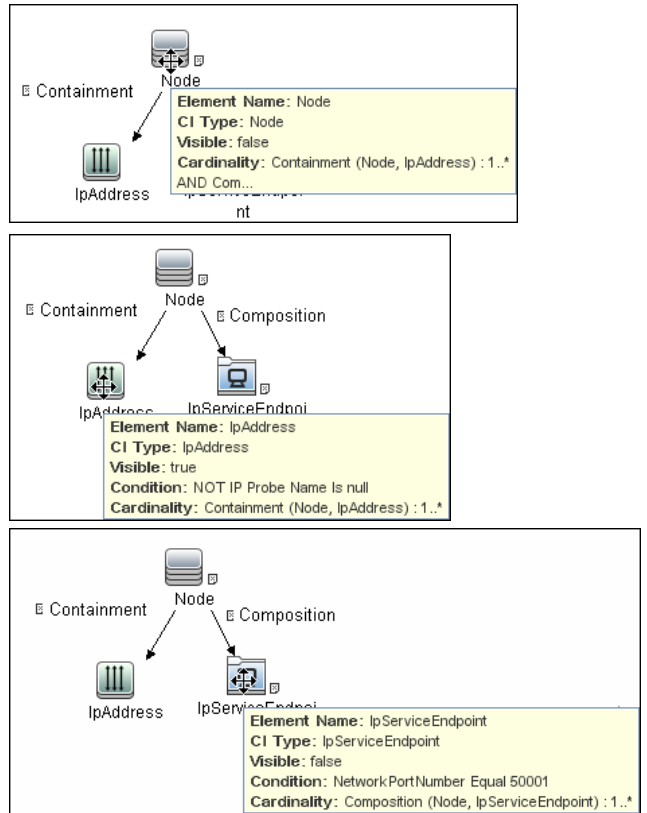
7 Trigger Query for the SIM Webservice Ports Job



8 Input Query for the SIM Webservice Ports Job



9 Trigger Query for the SIM Integration by WebServices Job



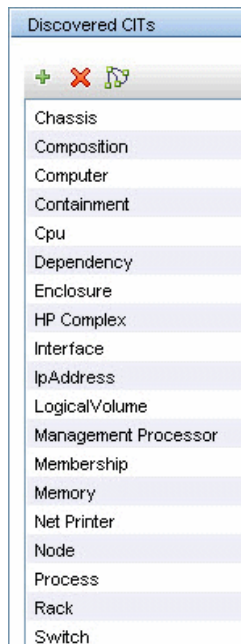
10 Discovery Workflow

- To discover the IP address of the HP SIM server, run the **Range IPs by ICMP** job (**Discovery Modules > Network – Basic**).
- To discover the Web service ports on the HP SIM server, run the **SIM WebService Ports** job (**Discovery Control Panel > Discovery Modules > Discovery-Based Product Integrations > Systems Insight Manager**). This job triggers on all **IpAddress** CIs in the CMDB and looks for port 50001 (the port at which HP SIM listens for Web service queries).

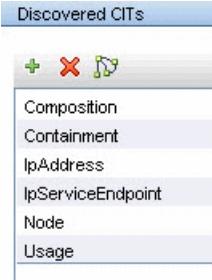
- c To discover HP SIM infrastructure, run the **SIM Integration by WebServices** job (**Discovery Control Panel > Discovery Modules > Discovery-Based Product Integrations > Systems Insight Manager**). This job triggers on results from the **SIM WebService Ports** job and retrieves data.

11 Discovered CITs – The SIM Integration by WebServices Job

To view discovered CITs, select a specific adapter in the Resources pane. For details, see "Discovered CITs Pane" in *HP Universal CMDB Data Flow Management Guide*.



12 Discovered CITs – The SIM Webservice Ports Job



Reference

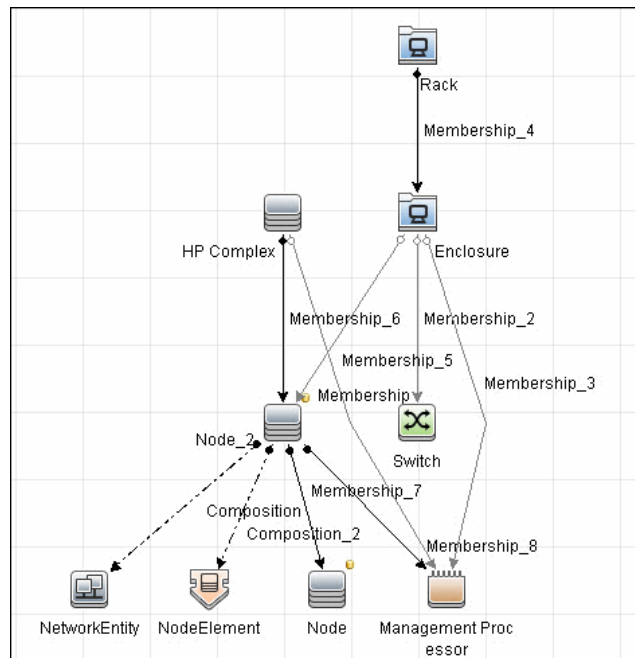
Instance Views

The package includes two adapter views that show all nodes and resources retrieved from HP SIM, as well as relationships between these nodes.

This section includes the following topics:

- "Host Infrastructure View" on page 723
- "Hosts and Resources from HP SIM" on page 724

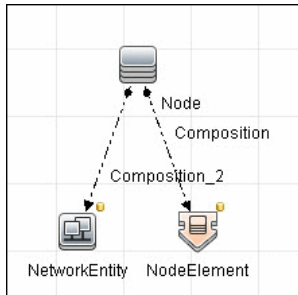
Host Infrastructure View



This view shows relationships between Chassis, Blade Enclosures, Servers, Workstations, Virtual Machine hosts to guests, and so on. This view also shows the interdependence between various nodes in an environment, to enable change management and correlation.

You can use this view, for example, to identify all the servers housed within a specific blade enclosure and all virtual machines running on servers within this blade enclosure. This enables analysis of the impact of shutting down a blade enclosure (say, for a firmware upgrade) on virtual machines. If UCMDB knows of services provided by these virtual machines and which business service these services are part of, it becomes possible to analyze the impact of a blade enclosure outage all the way to a business service.

Hosts and Resources from HP SIM



This view shows Node CIs retrieved from HP SIM with associated HostResource and NetworkResource CIs also retrieved from HP SIM.

Troubleshooting and Limitations

- ▶ If a **500: Internal Server Error** or **Out of Memory** error message appears in the Probe log files, enable chunking. For details, see "Enable Chunking (Optional)" on page 718.
- ▶ If there are multiple HP SIM servers in the environment and this discovery is used to integrate with all of them, you should create a new discovery job for each HP SIM server and schedule them to run separately. This is because the discovery uses XML files to process results from HP SIM, and running the discovery against multiple HP SIM servers simultaneously causes the XML files to be overwritten (because the file name is static).
- ▶ The **WebServices by URL** and **SIM Integration by WebServices** jobs use different versions of the **mxpartnerlib.jar**. Therefore, if you activate the **SIM Integration by WebServices** job, the **WebServices by URL** job cannot run (it fails with an exception), and vice versa (if you activate the **WebServices by URL** job, the **SIM Integration by WebServices** job cannot run).

To run both jobs, activate a job and obtain results. Then roll back the setup changes described in "Perform Setup on the Probe Machine" on page 717. Then run the second job.

45

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 728

Tasks

- ▶ Discover the ECC Storage Topology on page 729

Reference

- ▶ ECC Job SQL Queries on page 737
- ▶ Views on page 739
- ▶ Impact Analysis Rules on page 744
- ▶ Reports on page 747

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 729
- "Prerequisites" on page 730
- "Network and Protocols" on page 732
- "Discovery Workflow" on page 733
- "Discovery Adapter Parameters" on page 733
- "Discovered CIs" on page 734
- "ECC Integration Package" on page 735
- "Topology Map" on page 735

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 the *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 the *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.

- a** 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
- b** 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 733.
- c** Open the **WrapperEnv.conf** file with a text editor. The file is located in **C:\hp\UCMDB\DataFlowProbe\bin**.

- d** 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
```

- e** Append the following code to the end of the line beginning with `set.COMMON_CLASSPATH` (line ~87):

```
;%ORACLE_SSL_CLASSES%
```

- f** Save and close the file.
g 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 the *HP Universal CMDB Data Flow Management Guide*. For an explanation of a discovery job, see "Discovery Jobs" in the *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 the *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 737.

Tip: You can include the ECC job in the DFM schedule. For details, see "Discovery Scheduler Dialog Box" in the *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

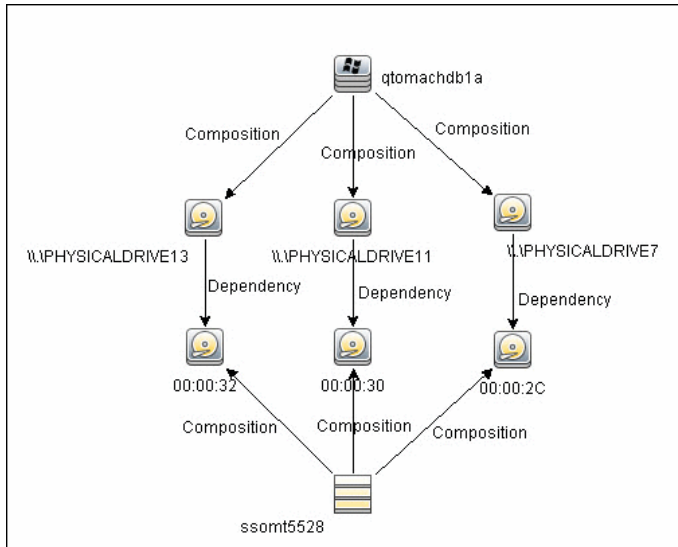
- CPU
- Containment
- Composition ([link](#))
- Dependency ([link](#))
- Fiber Channel Connect ([link](#))
- Fiber Channel HBA
- Fiber Channel Port
- Fiber Channel Switch
- Node
- IpAddress
- Logical Volume
- Membership ([link](#))
- Storage Array
- Storage Fabric
- Storage Processor
- Unix
- Windows

7 ECC Integration Package

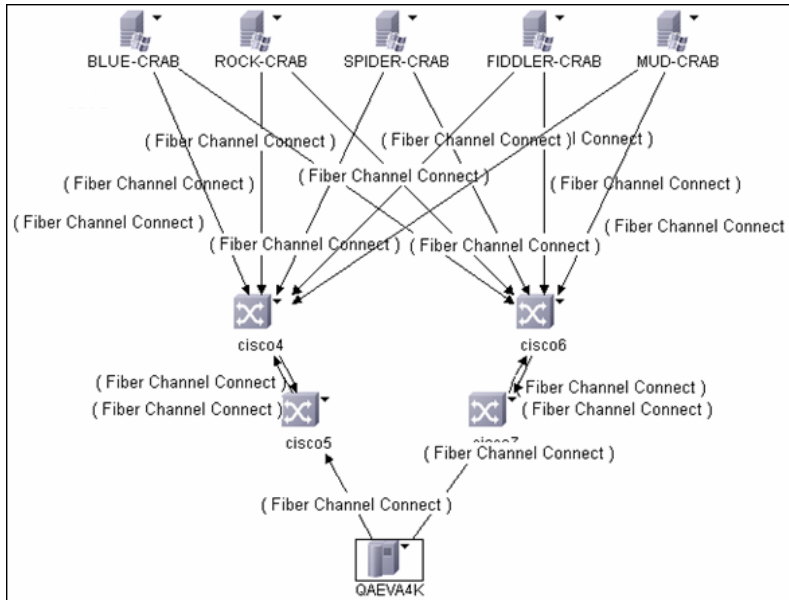
The integration includes the **ECC_Integration.zip** package, which contains the trigger TQL, DFM script, adapter, and job for ECC discovery. The DFM job uses the ECC Oracle database CI as the trigger.

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 732.
- 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 **Dependency** 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 the *HP Universal CMDB Administration Guide*.

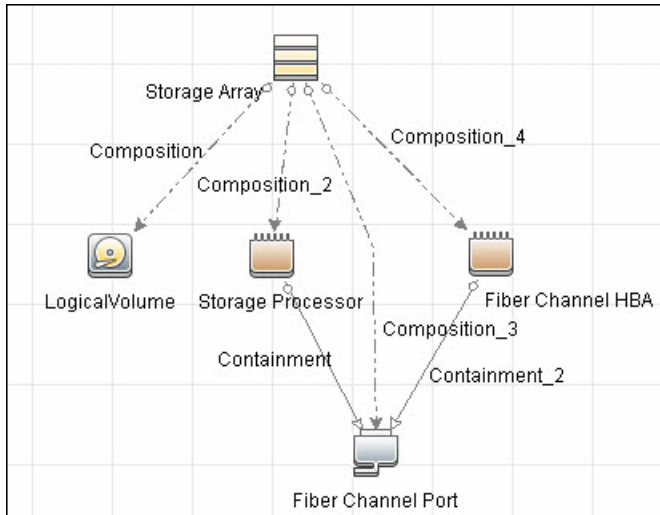
This section includes:

- "Storage Array Details" on page 740
- "FC Switch Details" on page 741
- "Storage Pool Details" on page 741
- "Host Storage Details" on page 742
- "SAN Topology" on page 743
- "Storage Topology" on page 743

Storage Array Details

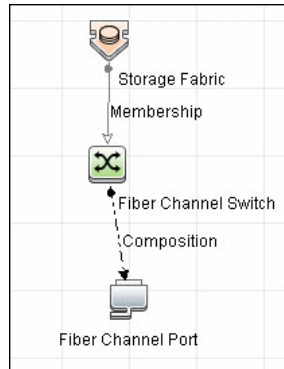
This view shows a Storage Array and its components including Logical Volumes, HBAs, Storage Processors, and Fiber Channel Ports. The view shows each component under its container Storage Array and groups Logical Volumes by CI Type.

Storage Array does not require all components in this view to be functional. Composition 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.



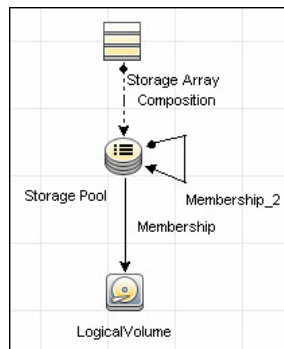
FC Switch Details

This view shows a Fiber Channel Switch and all connected Fiber Channel Ports.



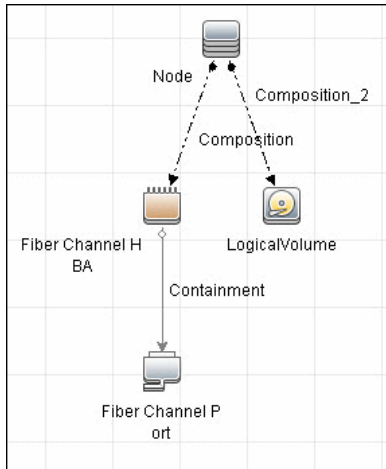
Storage Pool Details

This view shows Storage Pools with associated Storage Arrays and Logical Volumes.



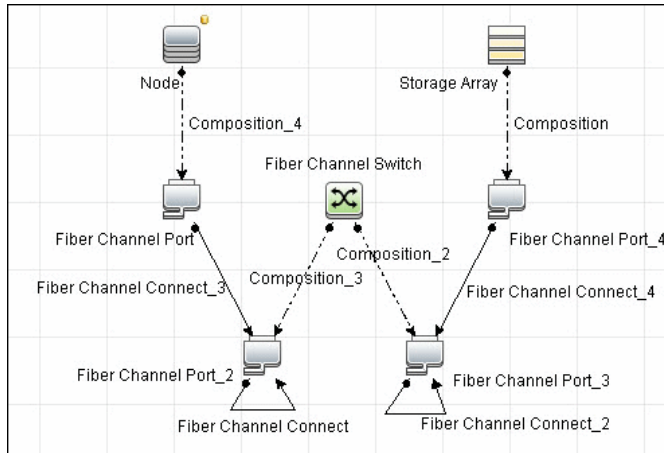
Host Storage Details

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.



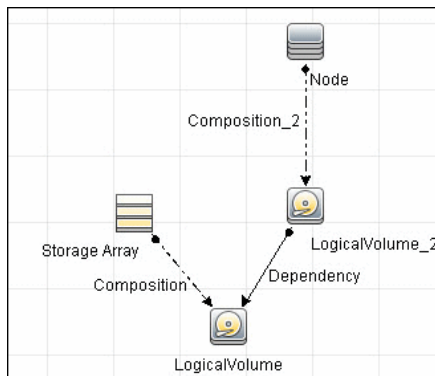
SAN Topology

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.



Impact Analysis Rules

The **Storage_Basic** package contains basic impact analysis rules to enable impact analysis and root cause analysis in UCMDB. These impact analysis rules are templates for more complex rules that you can define based on business needs.

All impact analysis rules fully propagate both Change and Operation events. For details on impact analysis, see "Impact Analysis Manager Page" and "Impact Analysis Manager Overview" in the *HP Universal CMDB Modeling Guide*.

To access the Storage_Basic package: **Administration > Package Manager**. For details, see "Package Manager" in the *HP Universal CMDB Administration Guide*.

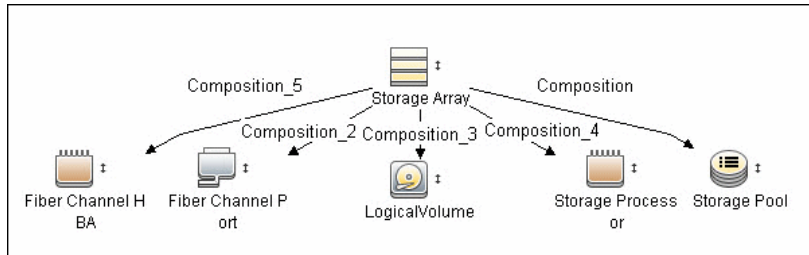
Note: Impact analysis events are not propagated to Fiber Channel Ports for performance reasons.

This section includes:

- ▶ "Storage Array Devices to Storage Array" on page 745
- ▶ "Host Devices to Host" on page 745
- ▶ "Logical Volume to Logical Volume" on page 745
- ▶ "FC Switch Devices to FC Switch" on page 746
- ▶ "FC Port to FC Port" on page 746

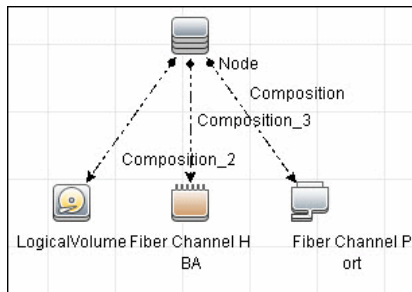
Storage Array Devices to Storage Array

This impact analysis rule propagates events between Logical Volumes, Storage Processors, Fiber Channel HBAs, and Storage Arrays.



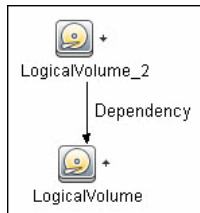
Host Devices to Host

This impact analysis rule propagates events between Fiber Channel HBAs and Hosts, and Logical Volumes on the Host.



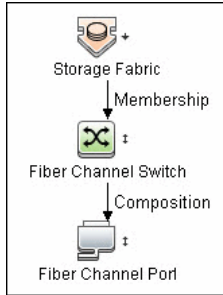
Logical Volume to Logical Volume

This impact analysis rule propagates events on a Logical Volume contained in a Storage Array to the dependent Logical Volume on the Host.



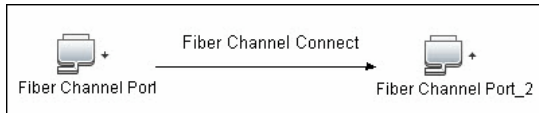
FC Switch Devices to FC Switch

This impact analysis 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 impact analysis 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 impact analysis 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.

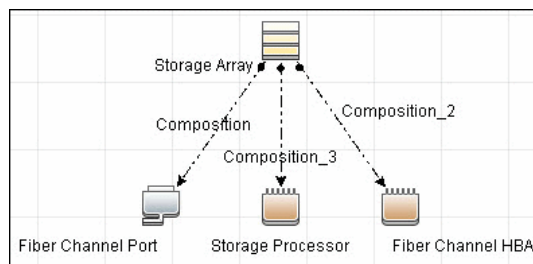
To access the Storage_Basic package: **Administration > Package Manager**. For details, see "Package Manager" in the *HP Universal CMDB Administration Guide*.

This section includes:

- "Storage Array Configuration" on page 747
- "Host Configuration" on page 748
- "Storage Array Dependency" on page 748
- "Host Storage Dependency" on page 749
- "Storage Pool Configuration" on page 749

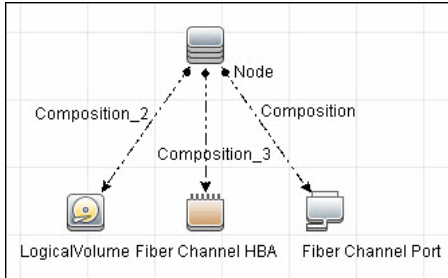
Storage Array Configuration

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.



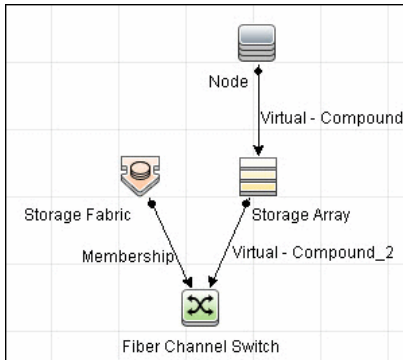
Host Configuration

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.



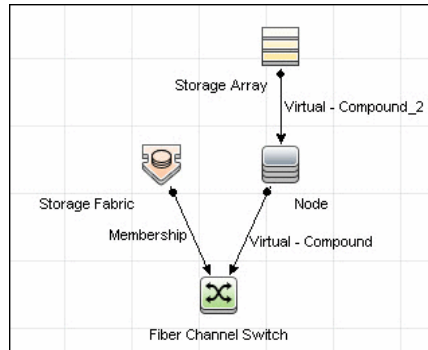
Storage Array Dependency

This report maps dependencies on a Storage Array. The report also displays information on switches connected to it.



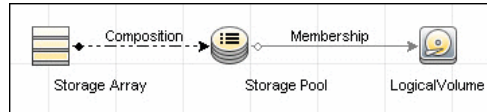
Host Storage Dependency

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.



46

Data Dependency and Mapping Inventory Integration with HP Universal CMDB

This chapter includes:

Concepts

- ▶ Overview on page 752
- ▶ DDMi Adapter on page 753

Tasks

- ▶ Populate the CMDB with Data from DDMi on page 755
- ▶ Federate Data with DDMi on page 758
- ▶ Customize the Integration Data Model in UCMDB on page 758

Reference

- ▶ DDMi Adapter Configuration Files on page 761

Troubleshooting and Limitations on page 762

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.

This section also includes:

- ▶ "Supported Versions" on page 752

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 755
- "Define a population job" on page 757
- "Run the population job" on page 757

1 Define the DDMi integration

a In UCMDB, 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 Point 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.

Name	Recommended Value	Description
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 "New Integration Job/Edit Integration Job 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 Universal CMDB 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 755.
- 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 *HP Universal CMDB 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 *HP Universal CMDB Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 762.

To add a new CI Type to the DDMi 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 *HP Universal CMDB 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 *HP Universal CMDB Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 762.

- 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 757 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 "Developing Generic Database Adapters" 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.

47

Microsoft SCCM/SMS Integration with HP Universal CMDB

This chapter includes:

Concepts

- SCCM/SMS Integration – Overview on page 764
- SMS Adapter on page 765

Tasks

- Populate the CMDB with Data from SCCM/SMS on page 767
- Federate Data with SCCM/SMS on page 771
- Customize the Integration Data Model in UCMDB on page 772

Reference

- SCCM/SMS Integration Package on page 774
- SMS Adapter Configuration Files on page 777

Troubleshooting and Limitations on page 778

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 765

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 767
- "Define a Population Job" on page 768
- "Run the population job" on page 770

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 Point 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 "New Integration Job/Edit Integration Job 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 Universal CMDB 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 767.
- 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 UCMDB

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 *HP Universal CMDB 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 *HP Universal CMDB Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 778.

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 *HP Universal CMDB 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 *HP Universal CMDB Modeling Guide*. For limitations on creating this TQL query, see "Troubleshooting and Limitations" on page 778.

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 774
- "SCCM/SMS Plug-in" on page 776
- "Reconciliation" on page 777

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	Represents number of the Windows service pack. SCCM/SMS DB: Service Pack 2 UCMDB: 2.0 Transformer: standard GenericEnumTransformer, mapped in the nt.nt_servicepack.transformer.xml file.
node	host_isdesktop	A Boolean value that determines whether a machine is a desktop or a server. SCCM/SMS DB: Workstation or Server UCMDB: true or false Transformer: standard GenericEnumTransformer, mapped in the node.host_isdesktop.transformer.xml file.

CMDB Class	Attribute	Transformation
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</p> <p>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>
node	host_osinstalltype	<p>Represents the Windows OS edition.</p> <p>SCCM/SMS DB. Microsoft Windows XP Professional</p> <p>UCMDB. Professional</p> <p>Transformer. Standard</p> <p>GenericEnumTransformer, mapped in the host.host_osinstalltype.transformer.xml file.</p> <p>Note: The same column in the SCCM/SMS database is mapped to two different UCMDB attributes, using different transformers.</p>

CMDB Class	Attribute	Transformation
disk device	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 UCMDDB 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 774.
- **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 "Developing Generic Database Adapters" 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.

48

Atrium Push Adapter

Note: This functionality is available as part of Content Pack 7.00 or later.

This chapter includes:

Concepts

- ▶ Overview on page 782

Tasks

- ▶ Integrate UCMDB with Remedy or Atrium on page 783

Reference

- ▶ Integration Mechanism on page 789
- ▶ Mapping Files on page 789
- ▶ **Troubleshooting and Limitations** on page 794

Concepts

Overview

HP Universal CMDB integrates with two BMC products:

- ▶ BMC Remedy Service Desk (Remedy)
- ▶ BMC Atrium CMDB (Atrium)

The integration adapter exports CIs and relationships from UCMDB to Remedy and Atrium.

The out-of-the-box integration does not transfer a specific list of CIs and relationships, but does enable you to replicate any CI or relationship from UCMDB to Remedy or Atrium.

For examples of enabling the integration with commonly used CIs and relationships, see "Discovery Workflow" on page 787.

Tasks

Integrate UCMDB with Remedy or Atrium

This task includes the following steps:

- "Supported Versions" on page 783
- "Prerequisites: Set Up Remedy Protocol" on page 783
- "Configure the Properties File" on page 783
- "Data Flow Probe Configuration" on page 785
- "Discovery Workflow" on page 787

1 Supported Versions

- BMC Remedy ARS: 7.0, 7.1, 7.5, 7.6
- BMC Atrium CMDB: 2.0, 2.1, 7.5, 7.6

2 Prerequisites: Set Up Remedy Protocol

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

3 Configure the Properties File

Configure the `push.properties` file: **Data Flow Management > Adapter Management > Resources > Packages > AtriumPushAdapter > Configuration Files > push.properties.**

- **pythonScript.name.** The name of the Python script that is invoked by this push adapter.
- **mappingFile.default.** The default XML mapping file used by mapping if a specific XML mapping file is not defined for an integration query. At least one default mapping file must be present in every adapter.

- **DebugMode.** If this value is set to **true**, the CI and relationships being pushed to Remedy/Atrium are also saved to XML files on the Data Flow Probe, under the following folder:
/discoveryResource/AtriumPushAdapter/work.
- **smartUpdateIgnoreFields.** A comma separated list of attributes (transferred from UCMDB to Atrium) that should **not** be used to check whether a CI has changed in Atrium. For example, as **updateTime** always changes, you would not want to update a CI in Atrium just because this attribute has changed.
- **sortCSVFields.** This parameter includes the TQL results of CSV aggregated fields that must always be sorted. When child attribute values are mapped and aggregated as CSV, the results are not sorted. This can trigger an update, even though nothing has changed in Atrium. To prevent an update, add here the CSV aggregated fields that must always be sorted.
- **testConnNameSpace.** Set this parameter to the **BMC NameSpace** being used for test connection purposes (for example, **BMC.CORE**).
- **testConnClass.** Set this parameter to the name of a BMC class, to query for connection test purposes (for example, **BMC_ComputerSystem**).

4 Data Flow Probe Configuration

- a** Copy the following JAR and DLL files from the BMC server to the following directory on the Data Flow Probe Server: **C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumPushAdapter**.

This directory is automatically created once the **AtriumPushAdapter** package is deployed on the UCMDB Server. If it is not present, ensure that the **AtriumPushAdapter** package has been correctly deployed on the UCMDB Server.

JAR Files	DLL Files
arapi75.jar	arapi75.dll
arutil75.jar	arencrypt75.dll
cmdbapi75.jar	arjni75.dll
commons-beanutils.jar	arrpc75.dll
commons-codec-1.3.jar	arutiljni75.dll
commons-collections-3.2.jar	arutil75.dll
commons-configuration-1.3.jar	arxmlutil75.dll
commons-digester-1.7.jar	cmdbapi75.dll
commons-lang-2.2.jar	cmdbjni75.dll
commons-logging-1.1.jar	icudt32.dll
log4j-1.2.14.jar	icuinbmc32.dll
oncrpc.jar	icuucbmc32.dll
spring.jar	Xalan-Cbmc_1_9.dll
	XalanMessagesbmc_1_9.DLL
	xerces-cbmc_2_6.dll
	xerces-depdombmc_2_6.dll

Note: The AR System Java API is forward and backward compatible with other versions of the AR System. For a complete compatibility matrix, refer to the "API Compatibility" section in the *BMC Remedy/Atrium Developer Reference Guide*.

- b** Edit the **WrapperGateway.conf** file (or **WrapperManager.conf** if the Probe Manager and Gateway are running in separate mode) in the following directory: **C:\hp\UCMDB\DataFlowProbe\bin**.

Add the following line after the **wrapper.java.library.path.2=%content_dll%** line:

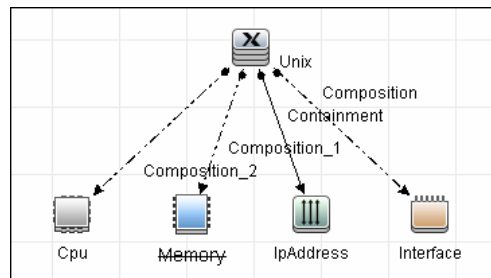
```
wrapper.java.library.path.3=%runtime%/probeManager/discoveryResources/AtriumPushAdapter
```

- c** Add the complete path to the Atrium DLL files (for example, **C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumPushAdapter**) to the Windows System Path on the Data Flow Probe machine.
- d** Restart the Data Flow Probe service.

5 Discovery Workflow

- a Configure Sync queries.** The CIs and relationships to be pushed to Remedy/Atrium have to be queried from UCMDB. Create queries (of type **Integration**) to query the CIs and relationships that have to be pushed to Remedy/Atrium.

An example of such a query (**atrium_push_sample_query**) is included with the Atrium package. To access the query: **Modeling > Modeling Studio > Root > Integration > Atrium**.



- b Create XML mapping files.** For every query created in the step above, create an XML mapping file with the same name as the integration query (the name must have the same case) in the following directory:

**C:\hp\UCMDB\UCMDBServer\runtime\fcmdb\CodeBase\
AtriumPushAdapter\mappings**

A sample mapping file (**atrium_push_sample_query.xml**) is provided out-of-the-box with the Atrium package.

For more details, see "Mapping Files" on page 789.

- c Create an Integration Point.** In the Adapter field, choose **AtriumPushAdapter**. Fill out the fields as follows:
- **Hostname/IP.** The host name or IP address of the BMC Remedy server.
 - **Port.** The port number of the BMC Remedy server.
 - **Credentials.** Open the **Choose Credentials** dialog box and select **Remedy Protocol**. Select the credentials to be used with this integration point. Click **OK**.
 - **Probe Name.** Select the Probe that should run this integration.

- Save the Integration Point.

For details, see "Integration Point Pane" in *HP Universal CMDB Data Flow Management Guide*.

- d Create a Job Definition.** For details, see "New Integration Job/Edit Integration Job Dialog Box" in *HP Universal CMDB Data Flow Management Guide*. Select the queries that will synchronize data between UCMDB and Remedy/Atrium.. Save the job definition and the integration point.
- e Invoke a full run of the job.** Click the **Run Full Job** button in the **Job Definition** tool bar. For details, see "Integration Jobs Pane" in *HP Universal CMDB Data Flow Management Guide*.

Reference

Integration Mechanism

Integration includes the following steps:

- 1 Queries UCMDB for CIs and relationships.** When an ad-hoc integration job is run in the Integration Studio, the integration process:
 - a** Receives the names of the integration queries that are defined in the job definition for that integration point.
 - b** Queries UCMDB for the results (new, updated, or deleted CIs and relationships) of these defined queries.
 - c** Applies the mapping transformation according to the pre-defined XML mapping files for every query.
 - d** Pushes the data to the Data Flow Probe.
- 2 Sends the data to BMC Remedy/Atrium.** On the Data Flow Probe, the integration process:
 - a** Receives the CI and relationship data sent from the UCMDB Server.
 - b** Connects to the BMC Remedy/Atrium server using the Java API.
 - c** Transfers the CIs and relationships.

Mapping Files

A mapping file is an XML file that defines which CIT or relationship in UCMDB is mapped to which CIT or relationship in the target data store.

Mapping files:

- Control which CITs and relationships are to be pushed.
- Control the attributes for the CITs and relationships that are to be mapped.
- Map attribute values from multiple CIs to one target CI.

- ▶ Map attributes of children CIs (those having a **containment** or **composition** relationship) to the parent CI in the target data store. For example:
 - ▶ Set a **Number of CPUs** value for a target **node** CI.
 - ▶ Set a **Total Memory** value for a target **node** CI.
- ▶ Map attributes of parent CIs (those having a **containment** or **composition** relationship) in the target data store CI. For example, in the Atrium target data store, set the value of a **Container Server** attribute on the **Installed Software** CIT by retrieving the value of the UCMDB **Installed Software** CI container node.

Mapping File Structure

Every mapping file has the following skeletal structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<integration>
  <info>
    <source ... .. />
    <target ... .. />
  </info>
  <source_ci_type name="...">
    <target_ci_type name="...">
      <targetprimarykey>
        <pkey>...</pkey>
      </targetprimarykey>
      <target_attribute name="..." datatype="..." >
        <map type="..." />
      </target_attribute>
    </target_ci_type>
  </source_ci_type>
</integration>
```

Note: An elipsis (...) signifies a configurable section.

Mapping File Elements

This section includes the following topics:

- "Main Parent Elements" on page 791
- "CI Type Mapping Elements" on page 791
- "Relationship Type Mapping Elements" on page 794

Main Parent Elements

- **<integration>**. The root element of the XML file. This element has no attributes.
- **<info>**. The source and target data stores being used, for example:

```
<info>
<source name="UCMDB" versions="9.x" vendor="HP" />
<target name="CACMDB" versions="12" vendor="CA" />
</info>
```

- **<targetcis>**. The element that encapsulates the mapping for all CI types.
- **<targetrelations>**. The element that encapsulates the mapping for all relationship types.

CI Type Mapping Elements

- **<source_ci_type>**. The element that defines a CI type of the source data store, for example:

```
<source_ci_type name="unix" mode="update_else_insert">
```

- **Attribute: name.** Defines the name of the source CI type.
- **Attribute: mode.** Defines the mode of the update in the target data store.
- **<target_ci_type>**. The element that defines the target CIT, for example:

```
<target_ci_type name="Hardware.Server.Unix">
```

- **Attribute: name.** Defines the name of the target CIT.

- **<targetprimarykey>**. The element that defines a list of all primary keys of the target CIT, for example:

```
<targetprimarykey>
  <pkey>host_key</pkey>
</targetprimarykey>
```

- **<target_attribute>**. This element that defines an attribute mapping from the source CI type to the target CI type attribute. Attribute mapping can be of the following types:

- **Constant**. This type enables setting a constant value on the target attribute:

```
<target_attribute name="DatasetId" datatype="char" length="127">
  <map type="constant" value="TOPO.DDM" />
</target_attribute>
```

- **Direct**. This type enables setting a direct value of a source data store attribute on the target data store:

```
<target_attribute name="Name" datatype="char" length="140">
  <map type="direct" source_attribute="host_hostname" />
</target_attribute>
```

- **Child Attribute**. This type enables retrieving attribute values of the source data store CI type children CIs and setting them on the target attribute. In the following example, the values of all the IpAddress CIs of a node CI are combined into a comma separated string and set on the target attribute **IPAddressList**:

```
<target_attribute name="IPAddressList" datatype="char">
  <map type="childattr">
    <aggregation type="csv"/>
    <source_child_ci_type name="ip_address" source_attribute="ip_address"/>
  </map>
</target_attribute>
```


- ▶ **Parent Attribute.** This type enables retrieving attribute values of the source data store CI type parent and setting it on the target attribute. In the following example, the id attribute value of the unix parent CIT is set to the target attribute **ParentChild**:

```
<target_attribute name="ParentChild" datatype="char">
  <map type="parentattr">
    <source_child_ci_type name="unix" source_attribute="id"/>
  </map>
</target_attribute>
```

- ▶ **Compound String.** This type enables the use of the above mapping types together to form more complex values for the target attribute, for example:

```
<target_attribute name="Bunch_O_Data" datatype="char" length="510"
  option="uppercase">
  <map type="compoundstring">
    <source_attribute name="name"/>
    <constant value="_UNIX_Server, IP="/>
    <childattr name="ip_address" source_attribute="ip_address"
      aggregation="csv"/>
    <constant value=", CPU="/>
    <childattr name="cpu" source_attribute="display_label" aggregation="csv"/>
  </map>
</target_attribute>
```

Relationship Type Mapping Elements

- **<link>**. The element that defines a relationship mapping from the source data store to a target data store, for example:

```
<link source_link_type="composition"
      target_link_type="BMC_HostedSystemComponents"
      source_ci_type_end1="unix"
      source_ci_type_end2="cpu"
      role1="Source"
      role2="Destination"
      mode="update_else_insert">
  <target_ci_type_end1 name="BMC_ComputerSystem"
    superclass="BMC_System" />
  <target_ci_type_end2 name="BMC_Processor"
    superclass="BMC_SystemComponent" />
  ... Relationship attribute mapping elements similar to the CI type attribute mapping
  elements ...
</link>
```

- **Attribute: source_link_type**. Defines the name of the source link.
- **Attribute: target_link_type**. Define the name of the target link.
- **Attribute: source_ci_type_end1**. The **End1** CI type of the source link.
- **Attribute: source_ci_type_end2**. The **End2** CI type of the source link.
- **<target_ci_type_end1>**. Used to specific the value of the target links end1 CI type
- **<target_ci_type_end2>**. Used to specific the value of the target links end2 CI type

Troubleshooting and Limitations

The integration mapping file enables the mapping only of concrete CI types and relationships to the CI types and relationships in BMC Remedy/Atrium. That is, a parent CIT cannot be used to map children CIs. For example, if **UCMDB Node** is mapped to **BMC_ComputerSystem**, any Node CIT of type **Unix** is not transferred. A mapping must be separately created for **Unix** to **BMC_ComputerSystem**.

Index

A

- Active Directory
 - discover 181
 - discover domain controllers, topology 183
- Active Directory Connection by LDAP job 185
- Active Directory Topology by LDAP job 186
- Actual State 622
- Actual State flow
 - configuration 623
 - queries 623
- adapter usage
 - federation 619
- adapters
 - configuration file for Service Manager 620
 - configuration file in ServiceCenter/Service Manager 625
 - deployment for ServiceCenter/Service Manager 634
 - deployment of ServiceDesk adapter 634
 - usage in ServiceCenter/Service Manager 619
- adsutil.vbs 43
- Alteon application switch by SNMP job 54
- Apache Tomcat by Shell job 45, 594, 597
- Application – SAP Solution Manager
 - module 238
- architecture 668
- attribute
 - adding to a CI type 651

B

- Books Online 16
- Bugzilla, Wordpress, MediaWiki 600

C

- CI type
 - adding 652
- Cisco CSS by SNMP job 55
- Class C IPs by ICMP job 254
- configuration file
 - for Service Manager adapter 620
 - for ServiceCenter/Service Manager adapter 625
- configuration files
 - SMS DB Adapter 777
- credential-less discovery 349
- CSV data
 - importing from external source 424

D

- data
 - import from external sources 422
- Data Flow Probe
 - copying files to a remote Windows machine 42
- data push
 - set up Service Manager 648
- data push flow 620
- Database Connections by Host Credentials
 - job 131
- Databases TCP Ports job 144, 150
 - jobs
 - Databases TCP Ports 167
- DB2 Topology by SQL job 144

- DB2 Universal Database Connection by SQL
 - job 144
- DDMi integration 751
 - overview 752
- discovery
 - applications 24
 - credential-less 349
 - IBM DB2 Server 143
 - IIS 603
 - JBoss 283
 - JBoss by Shell 287
 - Layer 2 383
 - localized versions 34
 - Microsoft Cluster Server 77
 - Microsoft Exchange Server 2007 199
 - Microsoft Load Balancing 81
 - Microsoft SQL Server 147
 - network - basic 327
 - operating systems 34
 - Oracle 165
 - SAP 229
 - SAP Solution Manager 238
 - Siebel 247
 - Solaris Zones 315
 - supported integration 35
 - Veritas Cluster Server 127
 - VMware 551
 - WebLogic 293
 - WebLogic by JMX 294
 - WebSphere 303
 - WebSphere by Shell 309
 - WebSphere Connections by JMX 305
- Discovery Tools 399
- diskinfo.exe 44
- documentation updates 20
- documentation, online 16

E

- ECC 727
 - discovering storage topology 729
 - overview 728
 - SQL queries for job 737
- ECC Integration by SQL job 733
- EMC Control Center
 - overview 728

- EMC Control Center integration 727
 - discovering storage topology 729
 - SQL queries for job 737
- error handling
 - Service Manager adapter 621
- Exchange_Server_2007_Discovery.ps1 45
- external sources
 - convert strings to numbers 422
 - importing data 422
 - importing data from 407
 - importing data, troubleshooting 428

F

- F5 BIG-IP LTM by SNMP job 54
- federation use cases 621
- File Monitor by Shell
 - limitation 400
- File Monitor by Shell job 45
- F-Secure
 - running on Windows for Host
 - Connection by Shell job 339

G

- general information 23, 37
- GetFileModificationDate.vbs 45
- getfilever.vbs 43

H

- Host Connection by Shell job 42, 158, 200, 235, 317, 597, 609
 - Windows running F-Secure 339
- Host Connection by Shell/WMI/SNMP job in ECC 731
- Host Connection by SNMP job 54, 317
- Host Connection by WMI job 194, 254, 317
- Host Fingerprint using nmap job 351
- Host Networking By SNMP job 386
- host resources
 - and application dependency, overview 372
 - and applications discovery 371
 - troubleshooting and limitations 381
 - workflow 375

- Host Resources and Applications by Shell job
 - 42, 43, 153, 200, 235, 244, 317, 372, 379, 597, 609
- Host Resources and Applications by Shell/WMI/SNMP job
 - in ECC 731
- Host Resources and Applications by SNMP job 317, 379
- Host Resources and Applications by WMI job 153, 317, 379, 381
- Host Resources and Applications module 371
- HP Service Manager
 - supported versions 620
- HP ServiceCenter
 - supported versions 620
- HP Software Support Web site 19
- HP Software Web site 20
- HP SIM
 - discover infrastructure 714
 - instance views 723
 - integration 709
 - troubleshooting and limitations 511, 550, 725
- Hyper-V 513

I

- IBM DB2 Server
 - discovery 143
- IHS Websphere Plugin by Shell job 45
- IIS
 - discovery 603
- IIS Applications by NTCMD job 43, 45, 604, 609
- import data from external sources 411
- Import from CSV File job 410, 411
- Import from Database job 410
- Import from Excel Workbook job 401, 402
 - troubleshooting 405
- Import from Properties job 410
- importing data
 - from external sources 407
 - troubleshooting 428
- installed software
 - reverting to previous discovery method 381

- integration
 - supported applications 35
- integration package
 - SMS 774
- integrations
 - with ECC 729
- IP Traffic by Network Data job 321

J

- J2EE JBoss by Shell job 286, 288
- J2EE TCP Ports job 307
- J2EE Weblogic by Shell job 45
- J2EE WebSphere by Shell job 45, 311
- J2EE WebSphere by Shell or JMX job 45, 307
- J2EE WebSphere Connections by JMX job 307
- JBoss
 - discovery 283
 - discovery by Shell 287
- JBoss by JMX job 286
- JBoss Connections by JMX job 286
- jobs
 - Alteon application switch by SNMP 54
 - Apache Tomcat by Shell 594, 597
 - Cisco CSS by SNMP 55
 - Class C IPs by ICMP 254
 - Database Connections by Host
 - Credentials 131
 - Databases TCP Ports 144, 150
 - DB2 Topology by SQL 144
 - DB2 Universal Database Connection by SQL 144
 - F5 BIG-IP LTM by SNMP 54
 - Host Connection by Shell 158, 200, 235, 317, 597, 609
 - Host Connection by SNMP 54, 317
 - Host Connection by WMI 194, 254, 317
 - Host Fingerprint using nmap 351
 - Host Networking By SNMP 386
 - Host Resources and Applications by Shell 153, 200, 235, 244, 317, 372, 379, 597, 609

Host Resources and Applications by
SNMP 317, 379

Host Resources and Applications by
WMI 153, 317, 379

IIS Applications by NTCMD 604, 609

Import from CSV File 410, 411

Import from Database 410

Import from Excel Workbook 401,
402

Import from Properties 410

IP Traffic by Network Data 321

J2EE JBoss by Shell 286, 288

J2EE TCP Ports 307

J2EE WebSphere by Shell 311

J2EE WebSphere by Shell or JMX 307

J2EE WebSphere Connections by JMX
307

JBoss by JMX 286

JBoss Connections by JMX 286

Layer2 Topology Bridge based by
SNMP 388

Layer2 Topology VLAN based by
SNMP 389

Microsoft Exchange Connection by
NTCMD 200

Microsoft Exchange Connection by
WMI 192, 194, 197

Microsoft Exchange Topology by
NTCMD 200

Microsoft Exchange Topology by
WMI 192, 194, 197

MS Cluster by NTCMD 78

MSSQL Server Connection by SQL 150

MSSQL Topology by SQL 150

MySQL by Shell 156, 158

Oracle Database Connection by SQL
167

Oracle Listeners by Shell 170

Oracle RAC Topology by Shell 170

Oracle Topology by SQL 167

Potential Servers by Network Data 322

Range IPs by ICMP 235, 317, 597

SAP ABAP Connection by SAP JCO
236

SAP ABAP Topology by SAP JCO 236

SAP Applications by SAP JCO 236

SAP ITS by NTCMD 236

SAP Java Topology by SAP JMX 244

SAP Solution Manager by SAP JCO
236

SAP System by Shell 236, 240, 244

SAP TCP Ports 239

SE Integration by SQL 694

Server Ports by Network Data 321

Servers by Network Data 319

Siebel DB by TTY 254, 260

Siebel DB by WMI and NTCMD 254

Siebel Web Applications by NTCMD
254

Siebel Web Applications by TTY 254

Software Element CF by Shell 379

TCP Ports 254

Veritas Cluster by Shell 129

VLAN ports by SNMP 387

VLANS by SNMP 387

VMware ESX Connection by VIM 556

VMware ESX Topology by VIM 556

VMware VirtualCenter Connection by
WMI and VIM 556

VMware VirtualCenter Topology by
VIM 556

WebLogic by Shell 298

WebServer Detection using TCP Ports
236, 254

WebServices by URL 604

junction.exe 46

K

Knowledge Base 19

L

Layer 2
discovery 383

Layer2 Topology Bridge based by SNMP job
388

Layer2 Topology VLAN based by SNMP job
389

M

- meminfo.exe 44
- Microsoft Cluster Server
 - discovery 77
- Microsoft Exchange Connection by NTCMD
 - job 45, 200
- Microsoft Exchange Connection by WMI job
 - 192, 194, 197
- Microsoft Exchange Server
 - discover 193
 - discover topology with Active Directory 202
 - overview 182, 192
- Microsoft Exchange Server 2007
 - discovery 199
 - package 201, 207
- Microsoft Exchange Topology by LDAP job
 - 207
- Microsoft Exchange Topology by NTCMD
 - job 45, 200
- Microsoft Exchange Topology by WMI job
 - 192, 194, 197
- Microsoft Internet Information Services (IIS)
 - 603
- Microsoft Message Queue
 - discovery 211
 - methodology 216
 - new entities 226
 - removed entities 227
 - task 212
- Microsoft Network Load Balancing
 - discovery 81
- Microsoft SCCM/SMS 763
- Microsoft SCCM/SMS integration 763
- Microsoft SQL Server
 - discovery 147
- MS Cluster by NTCMD job 78
- MSSQL Server Connection by SQL job 150
 - supporting named instances 151
- MSSQL Topology by SQL job 150
- multi-threading 620
- MySQL by Shell job 156, 158

N

- network - basic 327
- NLB 81
- NNMi integration 665
 - change management and impact
 - analysis 681, 683
 - connection protocol parameters 684
 - overview 666
 - run 670
 - troubleshooting and limitations 688
- NNMi-UCMDB Integration 678

O

- online documentation 16
- Online Help 17
- online resources 19
- Oracle
 - discovery 165
- Oracle Database Connection by SQL job 167
- Oracle Listeners by Shell job 170
- Oracle RAC Topology by Shell job 170
- Oracle TNSName by Shell job 45
- Oracle Topology by SQL job 167
- OS credentials
 - discovery for MS SQL Server 148
- Overview
 - DDMi integration 752

P

- Potential Servers by Network Data job 322
- processlist.exe 44
- push flow 620

Q

- queries
 - predefined 623

R

- Range IP by nmap job
 - jobs
 - Range IP by nmap 235

Index

- Range IPs by ICMP job 235, 317, 597
 - in ECC 731
- Readme 16
- reg_mam.exe 43
- remote Windows machine
 - deleting files from 41
- Running Software CF by Shell job 46

S

SAP

- discover ABAP 233
- discover Java 241
- discovery 229
- troubleshooting 245
- SAP ABAP Connection by SAP JCO job 236
- SAP ABAP discovery overview 230
- SAP ABAP Topology by SAP JCO job 236
- SAP Applications by SAP JCO job 236
- SAP ITS by NTCMD job 236
- SAP Java Topology by SAP JMX job 244
- SAP Profiles by Shell job 45
- SAP Solution Manager
 - discovery 238
- SAP Solution Manager by SAP JCO job 236
- SAP System By Shell job 45
- SAP System by Shell job 236, 240, 244
- SAP TCP Ports job 239
- SE 691
- SE Integration by SQL job 694
- Server Ports by Network Data job 321
- Servers by Network Data job 319
- Service Guard Cluster Topology by TTY job 45
- Service Manager adapter
 - flow and configuration 654
- ServiceCenter/Service Manager
 - adapter deployment 634
 - add attribute to CIT 640
- ServiceDesk adapter
 - deployment 634
- Siebel
 - discovery 247
- Siebel Application Server Configuration job 45
- Siebel DB by TTY job 254, 260

- Siebel DB by WMI and NTCMD job 254
- Siebel Web Applications by NTCMD job 254
- Siebel Web Applications by TTY job 254
- SMS
 - integration package 774
- SMS DB Adapter configuration files 777
- SMS integration
 - overview 764
- SMS supported versions 765
- SmsDbAdapter 765
- software
 - reverting to previous discovery method 381
- Software Element CF by Shell job 379
- Solaris Zones
 - discovery 315
- SQL Server
 - shallow discovery 152
 - shallow discovery overview 62, 98, 132, 153, 156, 358, 432, 472, 514, 530, 594, 710, 782
- SSL communication 647
- Storage Essentials (SE)
 - integration with Universal CMDB 692
- Storage Essentials integration 691
- supported versions
 - SMS 765

T

- TCP Ports job 254
- TempWmicBatchFile.bat
 - empty file created 42
- TQL queries
 - predefined 623
- Troubleshooting and Knowledge Base 19

U

- UCMDB
 - DDMi integration 751
- UDDI
 - discover processes 263
- updates, documentation 20

V

- Veritas Cluster by Shell job 46, 129
- Veritas Cluster Server
 - discovery 127
- VLAN ports by SNMP job 387
- VLANS by SNMP job 387
- VMware
 - discovery 551
- VMware ESX Connection by VIM job 556
- VMware ESX Topology by VIM job 556
- VMware VirtualCenter Connection by WMI and VIM job 556
- VMware VirtualCenter Topology by VIM job 556
- VMware VMotion
 - discover 571
- VMware VMotion Monitor by VIM job 573

W

- WebLogic
 - discovery 293
 - discovery by JMX 294
- WebLogic by Shell job 298
- Webserver by Shell job 46
- WebServer Detection using TCP Ports job 236, 254
- WebServices by URL job 604
- WebSphere
 - discovery 303
 - discovery by Shell 309
 - troubleshooting and limitations 312
- WebSphere Connections
 - discovery by JMX 305
- What's New 16

X

- xCmdSvc.exe
 - connecting to remote Windows machine 42
- Xen 577
 - discovery mechanism 587
 - discovery overview 578

