HP Universal CMDB

for the Windows and Solaris operating systems

Software Version: 8.02

HP Universal CMDB–HP Storage Essentials (SE) Integration Guide

Document Release Date: June 2009
Software Release Date: June 2009

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Trademark Notices

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

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

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

Unix® is a registered trademark of The Open Group.

Acknowledgements

For details, refer to the relevant product documentation.

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

HP Storage Essentials (SE) Integration with HP Universal CMDB.....7
Overview.....8
Run SE Discovery.....9
Run SE Discovery with a UCMDB Foundation License.....12
The Storage Essentials Integration Packages.....14
Discovered CITs.....14
Views.....24
Correlation Rules.....28
Reports.....30

HP Storage Essentials (SE) Integration with HP Universal CMDB

This document includes the main concepts, tasks, and reference information for integration of HP Storage Essentials (SE) with HP Universal CMDB (UCMDB).

This document includes:

Concepts

➤ Overview on page 8

Tasks

➤ Run SE Discovery on page 9

➤ Run SE Discovery with a UCMDB Foundation License on page 12

Reference

➤ The Storage Essentials Integration Packages on page 15

➤ Discovered CITs on page 15

➤ Views on page 25

➤ Correlation Rules on page 29

➤ Reports on page 31

Overview

This section describes how to integrate SE with UCMDB. Integration involves synchronizing the devices, topology, and hierarchy of a customer's 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.

You integrate SE with UCMDB using the Discovery and Dependency Mapping (DDM) application.

When you activate the **Integration – Storage Essentials** module, DDM retrieves data from the SE Oracle database and saves CIs to the UCMDB 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: DDM version 8.00 includes a module for discovering SE. No additional deployment is necessary.

Supported Versions

SE integration has been developed and tested on HP Universal CMDB versions 7.0 and 8.0 with SE version 6.x.

Run SE Discovery

This task includes the steps to run the SE/UCMDB integration jobs.

Note: If you are running UCMDB with a Foundation license, use the procedure in "Run SE Discovery with a UCMDB Foundation License" on page 12.

This task includes the following steps:

- "Prerequisites" on page 9
- "Activate the DDM Jobs" on page 9
- "Edit the TQL" on page 10
- "Activate the SE Integration by SQL Job" on page 10
- "Topology Map" on page 11

1 Prerequisites

The following VM installation prerequisites are required for SE integration:

- 4 GB memory.
- 50 GB hard drive space.

2 Activate the DDM Jobs

Run the following DDM jobs to discover the SE Oracle database:

- **Class C IPs by ICMP** or **Range IPs by ICMP** (under the Network – Basic module). Discovers the IP address of the Oracle database server. Not available with the Universal CMDB Foundation license.
- **Database TCP Ports** (under the Database – Oracle module). Discovers the Oracle TCP port and host instance on the IP address discovered in the previous DDM job. Not available with the Universal CMDB Foundation license.

3 Edit the TQL

Edit the **HPSE_OracleDB** TQL query (**Modeling** > **Query Manager** > **Discovery** > **HPSE_OracleDB**) and set a node condition on the instance name such that the TQL results only contain the HP SE database. For details on creating node conditions, see "Node/Relationship Properties Dialog Box" in *Model Management*.

4 Activate the SE Integration by SQL Job

SE Integration by SQL (under the Integration – Storage Essentials module). Discovers the Storage infrastructure.

The **SE Integration by SQL** DDM job uses a database CI as the trigger to run SQL queries against Oracle materialized views that are installed and maintained by SE in the Oracle database.

The SQL queries executed by the DDM job retrieve detailed information to build CIs and populate UCMDB.

This job can be activated manually or scheduled to run. For details, see "Discovery Scheduler Dialog Box" in *Discovery and Dependency Mapping*.

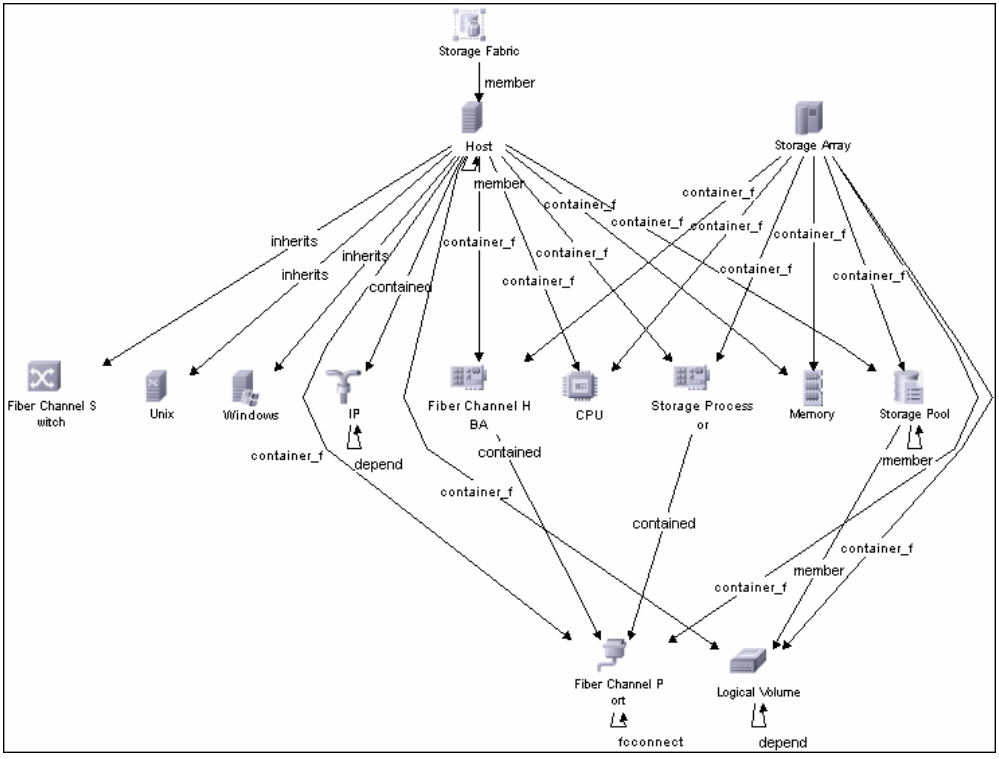
Note:

- For the Oracle server instance, it is recommended that you use the Oracle username **REPORT_USER**, since this user has privileges necessary to run SQL queries on the appiq_system tables.
- Since this DDM job queries Oracle materialized views, there is a chance that the views are being refreshed while the job is running, resulting in the display of an error message.

---

5 Topology Map

The following diagram illustrates the CITs discovered by the **SE Integration by SQL** job:



Run SE Discovery with a UCMDB Foundation License

If you are running UCMDB with a Foundation license, perform the following procedure to discover Storage Essentials components:

This task includes the following steps:

- "Prerequisites" on page 9
- "Activate the DDM Jobs" on page 12
- "Edit the TQL" on page 14
- "Topology Map" on page 14

1 Prerequisites

For details, see step 1 on page 9.

2 Activate the DDM Jobs

a Create a CI manually: **Admin** > **IT Universe Manager**. Click the **New CI** button to open the New CI dialog box. For details, see "New CI Dialog Box" in *Model Management*.

b In the Select CI Type pane, select **System** > **Host** > **Computer**. In the Define New CI Properties pane, in the **Host Key** box, enter the IP address of the SE server machine, followed by a space and **DefaultDomain**, for example, 1.2.3.4 DefaultDomain.

Click **Save** then **Close**.

c In the IT Universe Manager window, search for the CI that you created. UCMDB displays the CI in the CI Selector pane.

d Select the CI and click the **New Related CI** button to open the New Related CI dialog box. In the Select CI Type pane, select **All CI Types**.

e Select **System** > **Network Resource** > **Service Address**.

In the Define New CI Properties pane, under **Key properties**, in the **Service Address** box, enter the IP address of the SE server machine, followed by a colon and the Oracle port number, by default 1521, for example, 1.2.3.4:1521.

In the **Address Type** box, choose **tcp**.

Under **Properties inherited from class IT Universe**, locate the Name box. Type **oracle**.

Click **Relationship**.

f In the Select Relationship pane, choose **Container Link**.

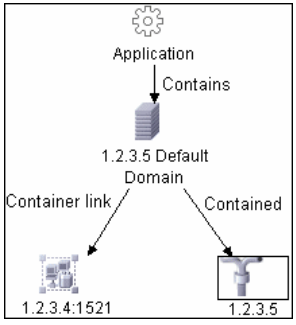
Click **Save** then **Close**.

g Repeat the steps for creating a related CI (make sure the new CI is selected in the CI Selector pane). This time, select **System** > **Network Resource** > **IP**. Under **Key properties**,in the **IP Address** box, enter the IP address of the SE server machine.

h Change the IP Domain Name value from **${DefaultDomain}** to the actual domain name (by default, **DefaultDomain**). In the Select Relationship pane, choose **Contained**.

Click **Save** then **Close**.

i Verify that the CIs were created successfully: Select the new CI. In the Active Directory Topology pane, right-click the new CI and select **Get Related CIs** > **From Database Without Filter**. The Related CIs from Database without Filter window should show a map of the related CIs that you created:



j You next define the IP address range of the SE server, needed by DDM: **DDM** > **Setup Discovery Probes** > **Domains and Probes** > **Default Domain** > **Probes**. Select the probe. Click the **Add IP** range button. Enter the range of the IP address. For details, see "Add/Edit IP Range Dialog Box" in *Discovery and Dependency Mapping*.

Note: The IP address range must have the following format:
start_ip_address – end_ip_address

k You now define the SQL protocol that is needed by DDM to discover the SE Oracle database: **DDM** > **Setup Discovery Probes** > **Domains and Probes** > **Default Domain** > **Credentials** > **SQL Protocol**. Click the **Add new connection** button to open the SQL Protocol Parameters dialog box. Fill in the details and save the parameters.

l Run the Oracle Connection by SQL job: **DDM** > **Run Discovery** > **Database – Oracle** > **Oracle Connection by SQL**.

m Run the SE Integration by SQL job: **DDM** > **Run Discovery** > **Integration – Storage Essentials** > **SE Integration by SQL**.

3 Edit the TQL

For details, see step 3 on page 10.

4 Topology Map

For details, see step 5 on page 11.

The Storage Essentials Integration Packages

The integration includes two UCMDB packages:

- **SE_Discovery.zip**. Contains the trigger TQL for SE discovery, discovery script, discovery pattern, and discovery job. The discovery pattern has no parameters and requires no configuration.
- **Storage_Basic.zip**. Contains the new CI Type definitions, views, reports, and correlation rules. This package is common to all Storage Management integration solutions.

Tip: You can include the SE job in the DDM schedule. For details, see "Discovery Scheduler Dialog Box" in *Discovery and Dependency Mapping*.

Discovered CITs

The following new CI types (CITs) in the UCMDB CI Type model represent SE 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 network resources.
- **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.

  The Storage_Basic.zip 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 UCMDB uses the IP address of a host as part of its primary key, this discovery pattern attempts to use an IP address from SE for this purpose. If an IP address is not available, the discovery 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.

The queries below also retrieve details on storage domains and storage fabrics.

```
SELECT switch.switchid, switch.switchname, switch.cimdomainid, switch.vendor,
switch.description, switch.appiq_last_contacted, switch.ip, switch.dns, switch.wwn,
switch.model, switch.serialnumber, switch.version, switch.switchstatus,
switch.switchstate, switch.switchrole, switch.fabricid, switch.fabricwwn,
switch.fabricname FROM appiq_system.mvc_switchsummaryvw switch WHERE
switch.status<>8
```

```
SELECT switch.availableports, switch.connectedports, switch.totalports FROM
appiq_system.mvc_swtichconfigvw switch WHERE switch.switchid = switchID from
above query
```

```
SELECT port.portid, port.portname, port.domainid, port.description, port.wwn,
port.connected_to_wwn, port.portstate, port.portstatus, port.port_speed,
port.max_speed, port.portnumber, port.scsiport, port.port_symbolic_name,
port.porttype, port.link_technology, port.trunkedstate FROM
appiq_system.mvc_portsummaryvw port WHERE port.containerid = switchID from
above query
```

Results from these queries populate a map as shown below:

- **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 Chassis CIT under network resources.

The Storage_Basic.zip package retrieves storage array details from the mvc_storagesystemsummaryvw view. The discovery pattern retrieves detailed information about Storage Processors and HBAs from the mvc_storageprocessorsummaryvw and mvc_cardsummaryvw tables, respectively.

For a variety of technical and policy-related reasons, the SE database may not be able to obtain IP address information about storage arrays. Since a storage array is a host as far as UCMDB is concerned, the discovery assumes that the serial number of a storage array is unique and uses this number 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, the discovery pattern 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, the discovery pattern queries the view by container ID for each storage array, each storage processor on a storage array, and each HBA on a storage array.

Finally, the discovery pattern retrieves detailed information about logical volumes on each storage array from the mvc_storagevolumesummaryvw view.

```
SELECT array.storagesystemid, array.storagesystemname, array.domainid,
array.vendor, array.description, array.ip, array.model, array.serialnumber,
array.version, array.storagesystemstatus, array.provider_tag FROM
appiq_system.mvc_storagesystemsummaryvw array WHERE array.status<>8
```

SELECT storageProcessor.systemprocessorid, storageProcessor.systemprocessorname, storageProcessor.domainid, storageProcessor.vendor, storageProcessor.description, storageProcessor.ip, storageProcessor.dns, storageProcessor.wwn, storageProcessor.model, storageProcessor.powermanagement, storageProcessor.serialnumber, storageProcessor.version, storageProcessor.processorstatus, storageProcessor.resetcapability, storageProcessor.roles, storageProcessor.providertag FROM appiq_system.mvc_storageprocessorsummaryvw storageProcessor WHERE storageProcessor.status<>8 AND storageProcessor.containerid = storageArrayID from above query

SELECT hba.cardid, hba.cardname, hba.cardtype, hba.domainid, hba.vendor, hba.description, hba.wwn, hba.model, hba.serialnumber, hba.version, hba.firmware, hba.driverversion FROM appiq_system.mvc_cardsummaryvw hba WHERE hba.status<>8 AND hba.containerid = storageArrayID from above query

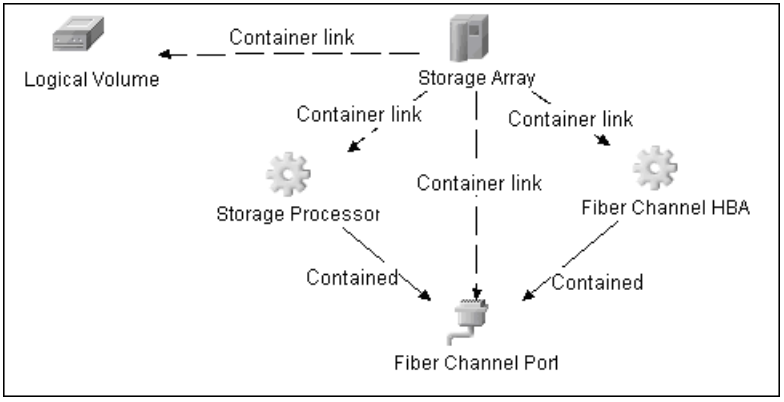
SELECT port.portid, port.portname, port.domainid, port.description, port.wwn, port.connected_to_wwn, port.portstate, port.portstatus, port.port_speed, port.max_speed, port.portnumber, port.scsiport, port.port_symbolic_name, port.porttype, port.link_technology, port.trunkedstate, port.containerid FROM appiq_system.mvc_portsummaryvw port WHERE port.status<>8 AND port.containerid = storageArrayID from above query

SELECT port.portid, port.portname, port.domainid, port.description, port.wwn, port.connected_to_wwn, port.portstate, port.portstatus, port.port_speed, port.max_speed, port.portnumber, port.scsiport, port.port_symbolic_name, port.porttype, port.link_technology, port.trunkedstate, port.containerid FROM appiq_system.mvc_portsummaryvw port WHERE port.status<>8 AND port.containerid IN (SELECT hba.cardid FROM appiq_system.mvc_storagesystemsummaryvw stor, appiq_system.mvc_cardsummaryvw hba WHERE hba.containerid = storageArrayID from above query)

```
SELECT port.portid, port.portname, port.domainid, port.description, port.wwn,
port.connected_to_wwn, port.portstate, port.portstatus, port.port_speed,
port.max_speed, port.portnumber, port.scsiport, port.port_symbolic_name,
port.porttype, port.link_technology, port.trunkedstate, port.containerid FROM
appiq_system.mvc_portsummaryvw port WHERE port.status<>8 AND
port.containerid IN (SELECT storageprocessor.systemprocessorid FROM
appiq_system.mvc_storagesystemsummaryvw stor,
appiq_system.mvc_storageprocessorsummaryvw storageprocessor WHERE
storageprocessor.containerid = storageArrayID from above query)
```

```
SELECT logicalVolume.storagevolumeid, logicalVolume.storagevolumename,
logicalVolume.domainid, logicalVolume.accesstype, logicalVolume.availability,
logicalVolume.statusinfo FROM appiq_system.mvc_storagevolumesummaryvw
logicalVolume WHERE logicalVolume.status<>8 AND
logicalVolume.storagesystemid = storageArrayID from above query
```

Results from these queries populate a map as shown below:

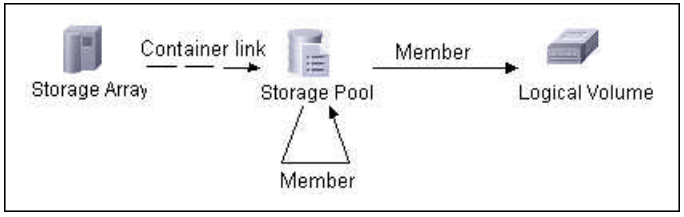


- **Storage Processor**. This CIT represents other storage devices such as SCSI controllers, and inherits from the Host Resource CIT. This CIT has change monitoring on parameters such as state, status, version, WWN, roles, power management, and serial number.
- **Storage Fabric**. This CIT inherits from Network Resource CIT and represents a storage fabric. This CIT has no change monitoring enabled.

➤ **Storage Pool**. Storage pool information is also collected from each storage array using the query below.

```
SELECT pool.storagepoolid, pool.storagepoolname, pool.storagepooldescription,
pool.parentpoolid, pool.cimpoolid, pool.pooltype, pool.storagecapabilityname,
pool.nosingleptoffailure, pool.defaultnosingleptoffailure, pool.mindataredundancy,
pool.maxdataredundancy, pool.minspindleredundancy, pool.maxspindleredundancy,
pool.default_spindle_redundancy, pool.storagecapabilitycommonname,
pool.storagecapabilitydescription, poolConfig.capacitytype, poolConfig.capacitynum,
poolConfig.exportedmb, poolConfig.unexportedmb, poolConfig.availablemb,
poolConfig.provisionedmb, poolConfig.totalmb FROM
appiq_system.mvc_storgaepoolsummaryvw pool,
appiq_system.mvc_storagepoolconfigvw poolConfig WHERE pool.status <> 8 AND
pool.storagesystemid = storageArrayID from above query AND pool.storagepoolid =
poolConfig.storagepoolid AND poolConfig.collectiontime IN (SELECT
MAX(collectiontime) FROM appiq_system.mvc_storagepoolconfigvw) ORDER BY
pool.parentpoolid DESC
```

Results from this query populate a map as shown below:



The following section includes

➤ "Host Details" on page 21

➤ "SAN Topology Details" on page 23

➤ "Storage Topology Details" on page 24

Host Details

The Storage_Basic.zip package retrieves host details from the mvc_hostsummaryvw view. The discovery pattern retrieves detailed information about HBAs from the mvc_cardsummaryvw view.

SE maintains information about operating systems, CPUs, memory, IP address, and DNS name on each host. UCMDB uses this information to create Host CIs of type UNIX or Windows, and adds CPU and Memory CIs for each host as available.

Since UCMDB uses the IP address of a host as part of its primary key, this discovery pattern attempts to use the IP address from SE for this purpose. If an IP address is not available, the pattern then attempts to resolve the host's IP address using a DNS name. If neither an IP address nor a DNS name is available, the discovery pattern 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 discovery pattern 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 the container ID as the key, the discovery pattern queries the view by container ID for each host, and each HBA on a host.

Finally, the discovery pattern 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 discovery pattern uses only the latest available information.

```
SELECT host.hostid, host.hostname, host.domainid, host.vendor, host.description,
host.ip, host.dns, host.model, host.version, host.os, host.totalphysicalmem,
host.numberprocessor FROM appiq_system.mvc_hostsummaryvw host WHERE
host.status<>8
```

```
SELECT hba.cardid, hba.cardname, hba.cardtype, hba.domainid, hba.vendor,
hba.description, hba.wwn, hba.model, hba.serialnumber, hba.version, hba.firmware,
hba.driverversion FROM appiq_system.mvc_cardsummaryvw hba WHERE
hba.status<>8 AND hba.containerid = hostID from above query
```

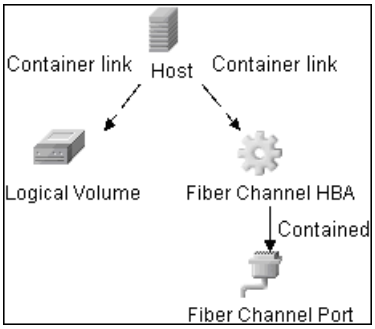
SELECT port.portid, port.portname, port.domainid, port.description, port.wwn, port.connected_to_wwn, port.portstate, port.portstatus, port.port_speed, port.max_speed, port.portnumber, port.scsiport, port.port_symbolic_name, port.porttype, port.link_technology, port.trunkedstate, port.containerid FROM appiq_system.mvc_portsummaryvw port WHERE port.status<>8 AND port.containerid = hostID from above query

SELECT port.portid, port.portname, port.domainid, port.description, port.wwn, port.connected_to_wwn, port.portstate, port.portstatus, port.port_speed, port.max_speed, port.portnumber, port.scsiport, port.port_symbolic_name, port.porttype, port.link_technology, port.trunkedstate, port.containerid FROM appiq_system.mvc_portsummaryvw port WHERE port.status<>8 AND port.containerid IN (SELECT hba.cardid FROM appiq_system.mvc_storagesystemsummaryvw stor, appiq_system.mvc_cardsummaryvw hba WHERE hba.containerid = hostID from above query)

SELECT logicalVolume.logicalvolumeid, logicalVolume.logicalvolumename, logicalVolume.domainid, logicalVolume.description, logicalVolume.deviceid, logicalVolume.filesystemtype, logicalVolume.share_name FROM appiq_system.mvc_hostvolumesummaryvw logicalVolume WHERE logicalVolume.status<>8 AND logicalVolume.hostid = hostID from above query

SELECT logicalVolume.total, logicalVolume.used, logicalVolume.free FROM appiq_system.mvc_hostcapacityvw logicalVolume WHERE LOWER(logicalVolume.capacitytype) = \'raw\' AND logicalVolume.volumeid = logicalVolumeID from above query AND logicalVolume.timestamp IN (SELECT MAX(lv.timestamp) FROM appiq_system.mvc_hostcapacityvw lv WHERE LOWER(lv.capacitytype) = \'raw\' AND lv.volumeid = logicalVolumeID from above query).

Results from these queries populate a map as shown below:

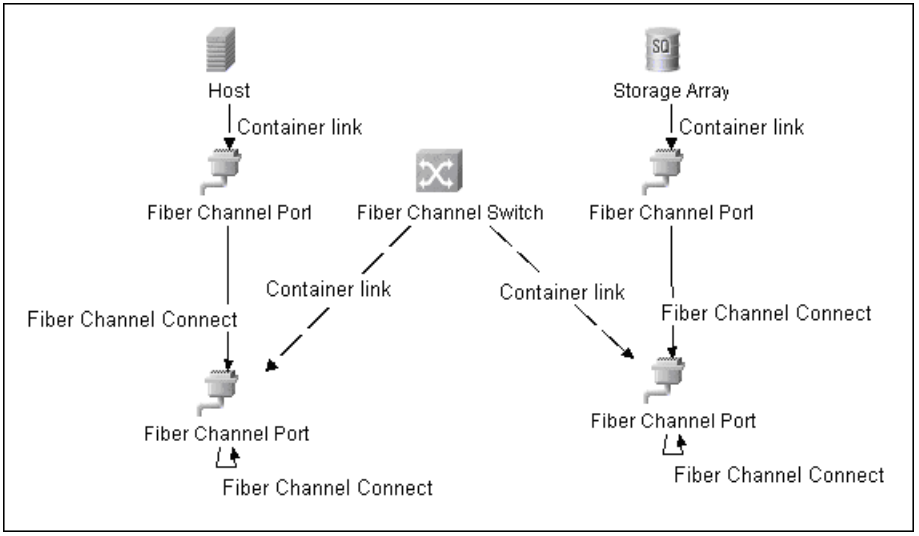


SAN Topology Details

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 to which each fiber channel port connects, and the Storage_Basic.zip package uses this list of WWNs to establish fiber channel connection links.

```
SELECT port.wwn, port.connected_to_wwn FROM appiq_system.mvc_portsummaryvw
port WHERE port.wwn IS NOT NULL AND port.connected_to_wwn IS NOT NULL
```

Results from this query populates a map as shown below:

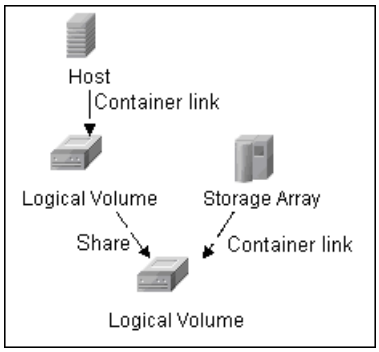


Storage Topology Details

Storage topology consists of relationships between logical volumes on a host and logical volumes on a storage array. The Storage_Basic.zip package uses multiple tables to identify this relationship as shown in the query below. This view is a simple summary of all of the above information.

```
SELECT DISTINCT hv.logicalvolumeid, sv.storagevolumeid FROM
appiq_system.mvc_hostsummaryvw h, appiq_system.mvc_pathvw ap,
appiq_system.mvc_subpathvw p, appiq_system.mvc_diskdrivesummaryvw ds,
appiq_system.mvc_hostvolumesummaryvw hv,
appiq_system.mvc_storgaepoolsummaryvw sp,
appiq_system.mvc_storagevolumesummaryvw sv,
appiq_system.mvc_storagevolumeports vp, appiq_system.mvc_protocolcontrollervw pc
WHERE ap.hostid=h.hostid AND ap.logicalvolumeid<>0 AND ap.ismountednum=1
AND p.pathid=ap.pathid AND p.diskdriveid=ds.diskdriveid AND
sv.storagevolumeid=p.storagevolumeid AND sp.storagepoolid=sv.poolid AND
sv.storagevolumeid=vp.storage_volume_id AND vp.port_id=pc.id AND
hv.logicalvolumeid=ap.logicalvolumeid
```

Results from this query populates a map as shown below:



Views

The **Storage_Basic.zip** package contains views that display common storage topologies. These basic views can be customized to suit the integrated SE applications.

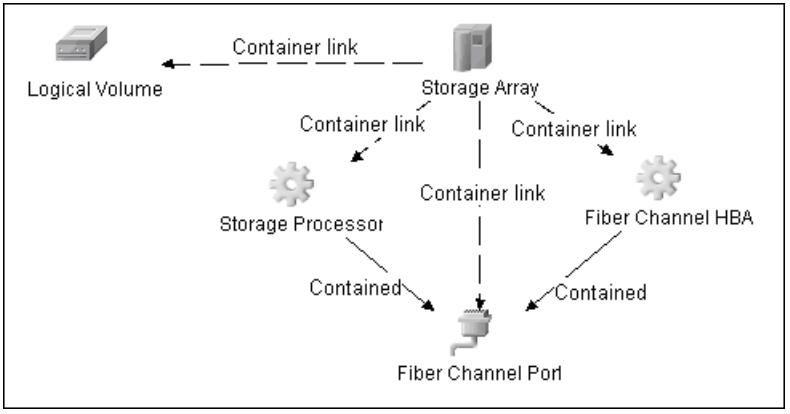
This section includes:

- "Storage Array Details" on page 26
- "FC Switch Details" on page 26
- "Storage Pool Details" on page 27
- "Host Storage Details" on page 27
- "SAN Topology Details" on page 28
- "Storage Topology Details" on page 28

Storage Array Details

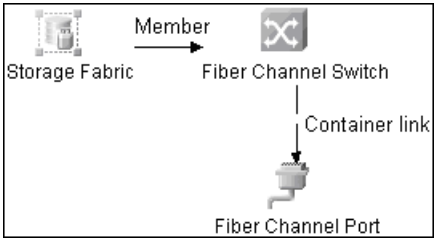
This view shows a storage array and all its components, including logical volumes, HBAs, storage processors, and fiber channel ports. The view shows each component under its container storage array, and groups logical volumes by CIT for clarity.

A storage array does not require all components in this view to be functional. Container links stemming from the storage array have a cardinality of zero-to-many. The view may show storage arrays whether or not there are logical volumes or storage processors.



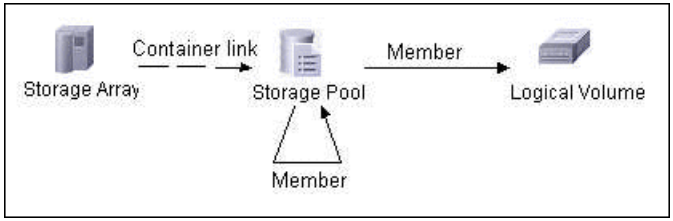
FC Switch Details

This view shows a fiber channel switch and all connected fiber channel ports. For clarity, the view shows Fiber Channel Ports under their container switch.



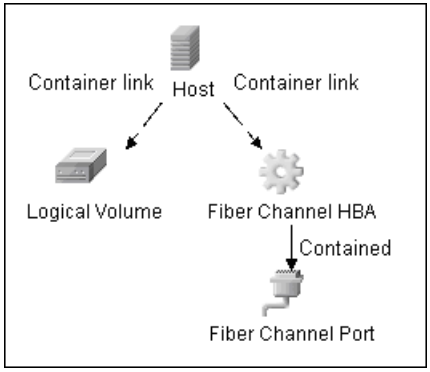
Storage Pool Details

This view shows Storage Pools with associated storage arrays and logical volumes.



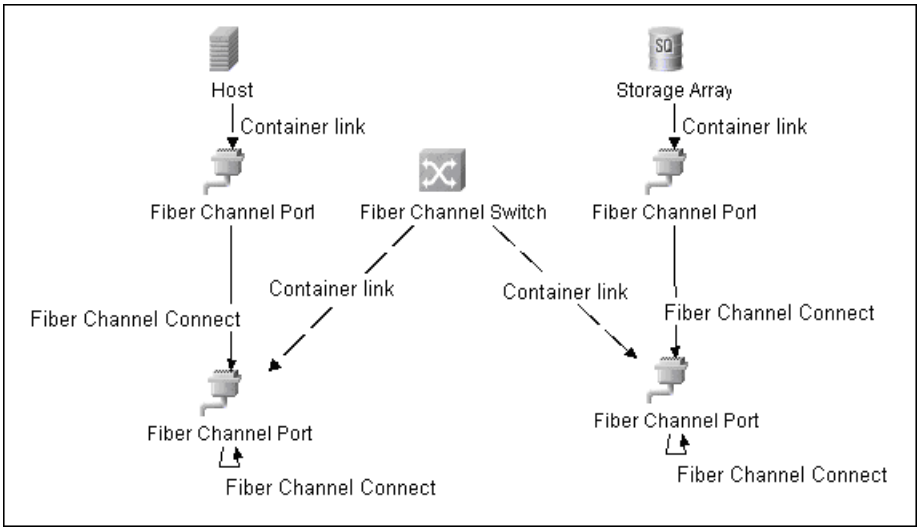
Host Storage Details

This view shows only Hosts that contain a fiber channel HBA or a logical volume. This keeps this view storage-specific and prevent hosts discovered by other UCMDB discovery jobs from appearing. The view shows logical volumes and fiber channel HBAs under their container host, and groups them by CIT for clarity.



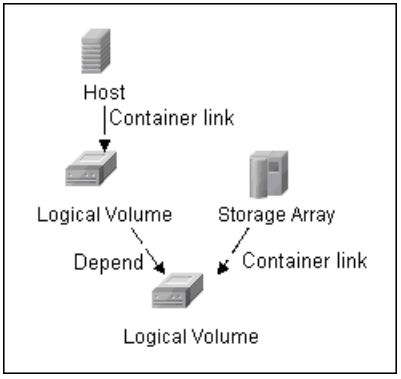
SAN Topology Details

This view maps physical connections between storage arrays, fiber channel switches and hosts. The view shows fiber channel ports below their respective containers, and groups the fiber channel connect relationship by CIT to prevent multiple relationships between the same nodes from appearing on the top layer.



Storage Topology Details

This view maps logical dependencies between logical volumes on hosts and logical volumes on storage arrays. There is no folding in this view.



Correlation Rules

The **Storage_Basic.zip** package contains basic correlation rules to enable impact and root cause analysis features in UCMDB. These correlation rules are templates for more complex rules that you can define based on business needs.

All correlation rules fully propagate both change and operation events. For details on impact analysis, see "Correlation Manager Overview" in *Model Management*.

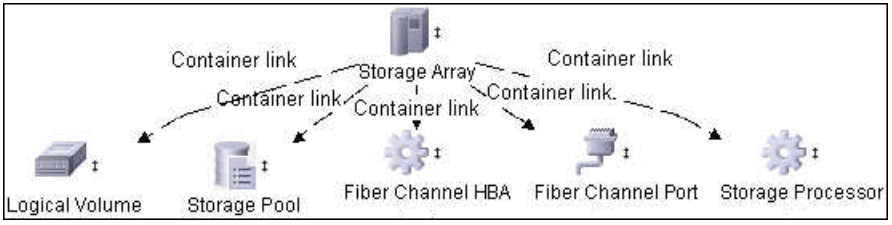
Correlation events are not propagated to fiber channel ports for performance reasons, since there can be many events. For details, see "Storage Array Devices to Storage Array" on page 29, "Host Devices to Host" on page 30, "Logical Volume to Logical Volume" on page 30, and "FC Switch Devices to FC Switch" on page 30.

This section includes:

- "Storage Array Devices to Storage Array" on page 29
- "Host Devices to Host" on page 30
- "Logical Volume to Logical Volume" on page 30
- "FC Switch Devices to FC Switch" on page 30
- "FC Port to FC Port" on page 31

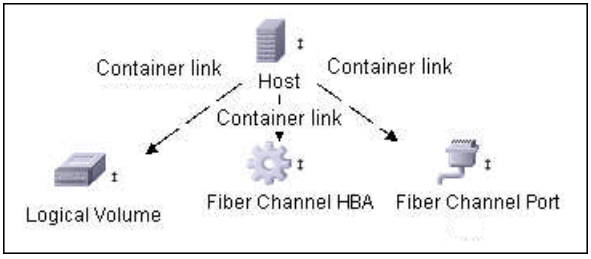
Storage Array Devices to Storage Array

This correlation rule propagates events between logical volumes, storage processors, fiber channel HBAs, and storage arrays.



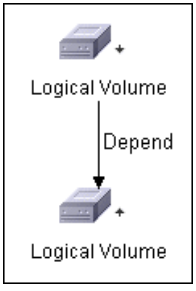
Host Devices to Host

This correlation rule propagates events between fiber channel HBAs, hosts, and logical volumes on the host.



Logical Volume to Logical Volume

This correlation rule propagates events on a logical volume contained in a storage array to the dependent logical volume on the host.

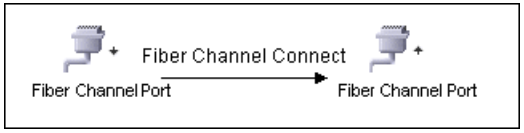


FC Switch Devices to FC Switch

This correlation rule propagates events from a fiber channel port to and from a switch, and to the associated storage fabric.

FC Port to FC Port

This correlation rule propagates events on a fiber channel port to another connected channel port.



For example, consider an HBA on a storage array going down. The event propagates from the HBA to the storage array and the logical volumes on the array because of the storage devices to storage array rule. The correlation event on the logical volume then propagates to other dependent logical volumes through the logical volume to logical volume rule. Hosts using those dependent logical volumes see the event next because of the host devices to host rule. Depending on business needs, you can define correlation rules to propagate events from these hosts to applications, business services, lines of business, and so on. This enables end-to-end mapping and impact analysis using UCMDB.

Reports

The **Storage_Basic.zip** 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 UCMDB. For details see "Storage Array Configuration" on page 32, "Host Configuration" on page 32, "Storage Array Dependency" on page 33, and "Host Storage Dependency" on page 33.

This section includes:

➤ "Storage Array Configuration" on page 32

➤ "Host Configuration" on page 32

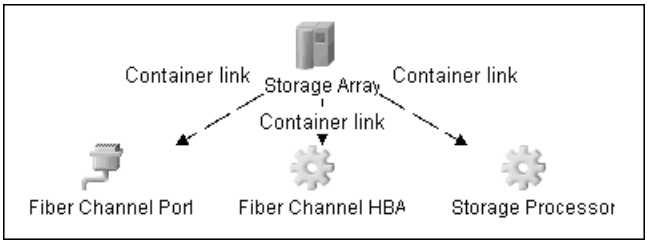
➤ "Storage Array Dependency" on page 33

➤ "Host Storage Dependency" on page 33

➤ "Storage Pool Configuration" on page 34

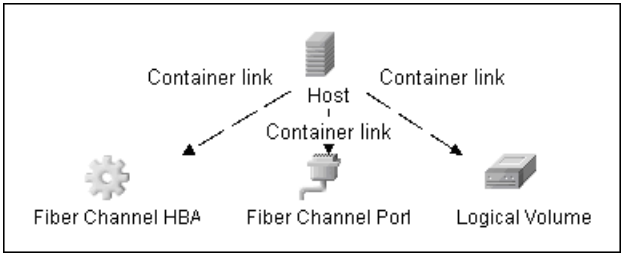
Storage Array Configuration

This report shows detailed information about storage arrays and their 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.



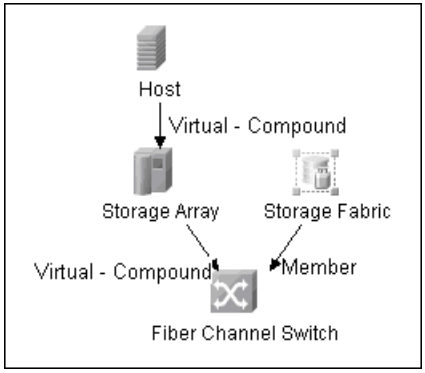
Host Configuration

This report shows detailed information about 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.



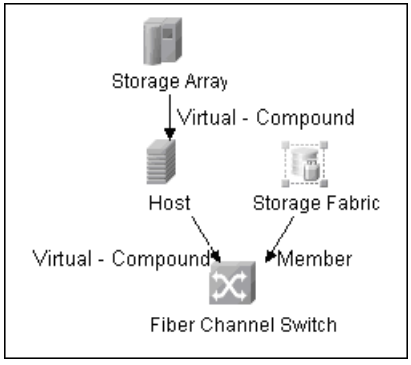
Storage Array Dependency

This report is an attempt to map dependencies on a storage array. The report also displays information about switches connected to the storage array.



Host Storage Dependency

This report shows detailed information about a host's storage infrastructure dependencies. The report lists hosts and their dependent components.



Storage Pool Configuration

This report shows detailed information about storage pool configuration.