

HP Service Manager Software

for the Windows® and Unix® operating systems

Software Version: 7.11

HP Universal CMDB to HP Service Manager Integration Guide

Document Release Date: July 2009

Software Release Date: July 2009



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 1994–2009 Hewlett-Packard Development Company, L.P.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Smack software copyright © Jive Software, 1998-2004. SVG Viewer, Mozilla JavaScript-C (SpiderMonkey), and Rhino software Copyright © 1998-2004 The Mozilla Organization. This product includes software developed by the OpenSSL Project for use in the OpenSSL toolkit. (<http://www.openssl.org>). OpenSSL software copyright 1998-2005 The OpenSSL Project. All rights reserved. This project includes software developed by the MX4J project (<http://mx4j.sourceforge.net>). MX4J software copyright © 2001-2004 MX4J Team. All rights reserved. JFreeChart software © 2000-2004, Object Refinery Limited. All rights reserved. JDOM software copyright © 2000 Brett McLaughlin, Jason Hunter. All rights reserved. LDAP, OpenLDAP, and the Netscape Directory SDK Copyright © 1995-2004 Sun Microsystems, Inc. Japanese Morphological Analyzer © 2004 Basis Technology Corp. The Sentry Spelling-Checker Engine Copyright © 2000 Wintertree Software Inc. Spell Checker copyright © 1995-2004 Wintertree Software Inc. CoolMenu software copyright © 2001 Thomas Brattli. All rights reserved. Coroutine Software for Java owned by Neva Object Technology, Inc. and is protected by US and international copyright law. Crystal Reports Pro and Crystal RTE software © 2001 Crystal Decisions, Inc., All rights reserved. Eclipse software © Copyright 2000, 2004 IBM Corporation and others. All rights reserved. Copyright 2001-2004 Kiran Kaja and Robert A. van Engelen, Genivia Inc. All rights reserved. Xtree copyright 2004 Emil A. Eklund. This product includes software developed by the Indiana University Extreme! Lab (<<http://www.extreme.indiana.edu/>>). Portions copyright © Daniel G. Hyans, 1998. cbg.editor Eclipse plug-in copyright © 2002, Chris Grindstaff. Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are copyright © 2001-2004 Robert A. van Engelen, Genivia Inc. All Rights Reserved. Copyright © 1991-2005 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Trademark Notices

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

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

UNIX® is a registered trademark of The Open Group.

Documentation Updates

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:

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

Contents

1	Introduction	7
	Purpose of the integration	7
	Who should read this guide?	8
	What information is in HP Universal CMDB?	8
	What information is stored in HP Service Manager?	8
2	Integration setup	11
	Integration requirements	12
	Integration setup overview	13
	Install the Service Manager integration package	14
	Add an RMI adapter data store	14
	Add a new Changes adapter data store	16
	Add a Service Desk adapter data store	19
	HP Service Manager setup overview	23
	Add the UCMDB connection information	23
	Edit the Process Definition record to support a UCMDB 7.52 system	24
	Add reconciliation rules	25
	Create a DEM reconciliation rule	26
	Add Discovery Event Manager rules	27
	Transfer UCMDB CIs to Service Manager	29
	Define a Changes replication job	29
	Define an RMI replication job	31
	Schedule replication jobs	33
3	Tailoring the integration	35
	Integration architecture	35
	Integration class model	35
	Integration TQL queries	39
	TQL query requirements	43

Service Manager web services	43
Managed fields	44
Service Manager reconciliation rules	49
Performance implications	50
Dependence on DEM rules	50
Service Manager Discovery Event Manager rules	50
Change the conditions under which a DEM rule runs	50
Change the action the DEM rule takes	52
Update the list of managed fields for a CI type	52
Create custom JavaScript to open change or incident records	52
Making the integration components visible in UCMDB	55
Change the Link Root setting	55
Change the Object Root setting	57
Tailoring options	59
Add a CI attribute to the UCMDB class model	60
Enable calculation for the CI attribute	62
Enable the Change Monitored option for CI attributes	64
Map the CI attribute to a web service object	66
Reload the code base for the ServiceDeskAdapter	69
Create a web service object to support the CI attribute	71
Add a managed field to support the CI attribute	72
Add a CI Type to the integration	76
Add a CI type to the UCMDB class model	77
Create a TQL query to synchronize the CI type	81
Verify all relationships in the TQL query have the tracking qualifier	85
Enable calculation for the CI type's attributes	86
Map the CI type's TQL query to an XSL transformation file	89
Map the CI type's attributes to web service objects	92
Reload the code base for the ServiceDeskAdapter	97
Add custom TQL queries to the integration Changes Adapter	99
Add custom TQL queries to integration replication jobs	100
Create web service objects to support the CI type	101
Add managed fields to support the CI type	103

1 Introduction

Purpose of the integration

An integration between HP Universal CMDB (UCMDB) and HP Service Manager enables you to share information about the actual state of a configuration item (CI) between your UCMDB system and a Service Manager system. Any organization that wants to implement the best practices Configuration Management and Change Management ITIL processes can use this integration to verify that CIs actually have the attribute values the organization has agreed to support.

You can use this integration to automate the creation of Service Manager change or incident records to update or rollback CIs that have unexpected attribute values. Service Manager allows you to programmatically define what actions you want to take whenever a CI's actual state does not match the expected state as defined in the CI record.

The integration offers several different ways for users to view CI actual state information:

- By default, the integration automatically updates the managed fields of Service Manager CI records as part of the regular UCMDB synchronization schedule. You can choose the option to configure the integration to automatically create change or incident records instead.
- A Service Manager user can view the current actual state of a CI by looking at the Actual State tab in the CI record. When you click on the Actual State tab, Service Manager makes a web services request to UCMDB and displays all CI attributes the request returns. Service Manager only makes the web service call when you click on the tab.
- A Service Manager user can use the **View in UCMDB** option to log in to the UCMDB system and view the current CI attributes from UCMDB. The Service Manager user must have a valid UCMDB user name and password to log in to the UCMDB system.

Who should read this guide?

This guide is intended for a system implementer or system administrator who will be establishing and maintaining a connection between the HP Universal CMDB and | HP Service Manager systems. This guide assumes that you have administrative access to both systems. The procedures in this guide may duplicate information available in your UCMDB and Service Manager help systems but is provided here for convenience.

What information is in HP Universal CMDB?

Your UCMDB system stores the actual state of CIs and CI relationships as CI attributes. Typically, UCMDB uses one or more discovery mechanisms (feeders) to automatically detect CI attribute values. The UCMDB to Service Manager integration only uses a subset of the CI attributes available in a UCMDB system.

For more information, see [Tailoring the integration](#) on page 35.

What information is stored in HP Service Manager?

Your Service Manager system stores the managed or expected state of CIs and CI relationships as attribute values in a CI record. To be part of the integration, a CI attribute in your UCMDB system must map to a managed field in the Service Manager CI record. You can add, remove, or update the managed fields that are part of the integration by tailoring the Service Manager web services that manage the integration.

Service Manager runs according to a set of rules that define what actions you want the system to take whenever a CI's actual state does not match the expected state as defined in the CI record. You define these rules from the Discovery Event Manager (DEM) in Service Manager where you can do the following:

- Automatically update a CI record to match the attribute values listed in the actual state. (This is the default behavior.)
- Automatically create a change record to review the differences between the actual state and the managed state

- Automatically create an incident record to review the differences between the actual state and the managed state

2 Integration setup

To assist you in setting up the integration between HP Service Manager and HP UCMDB, this chapter contains the following topics:

- [Integration requirements](#) on page 12
- [Integration setup overview](#) on page 13
- [HP Service Manager setup overview](#) on page 23
- [Transfer UCMDB CIs to Service Manager](#) on page 29

Integration requirements

You must setup the following required components to establish an integration between UCMDB and Service Manager.

- HP Universal CMDB 7.52 or higher installation
 - Configure a synchronization between a UCMDB feeder and UCMDB to discover the following CI attributes:
 - DDM (this is the UCMDB default discovery feeder mechanism)
 - DDMi (this is an optional integration to an external discovery feeder mechanism)
 - Configure a synchronization between UCMDB and Service Manager to send CI attributes:
 - Define a Service Desk target adapter to send CI attributes to Service Manager (required for all integrations)
 - Define an RMI source adapter to synchronize compound relations (required for DDMi only)
 - Define a Changes source adapter to synchronize CI changes (required for DDMi only)
- HP Service Manager 7.10 or higher installation
 - Add the UCMDB URL to the System Information Record
- Network connection between the HP Universal CMDB and HP Service Manager systems.

For instructions on installing and configuring your systems, see the UCMDB and Service Manager documentation.

Integration setup overview

The integration requires setup on both the UCMDB and Service Manager systems.

Task 1: Set up the UCMDB system.

See [HP Universal CMDB setup overview](#) on page 13.

Task 2: Set up the Service Manager system.

See [HP Service Manager setup overview](#) on page 23.

Task 3: Run the UCMDB replication jobs to transfer CIs to Service Manager.

See [Transfer UCMDB CIs to Service Manager](#) on page 29.

HP Universal CMDB setup overview

You must complete the following tasks from your UCMDB system to support the integration.

Task 1: Install the Service Manager integration package.

See [Install the Service Manager integration package](#) on page 14.

Task 2: Add an RMI adapter data store between UCMDB and the source feeder.

See [Add an RMI adapter data store](#) on page 14.

Task 3: Add a new Changes adapter data store between UCMDB and the source feeder.




See [Add a new Changes adapter data store](#) on page 16.

Task 4: Add a Service Desk adapter data store between UCMDB and Service Manager.

See [Add a Service Desk adapter data store](#) on page 19.


Install the Service Manager integration package

The Service Manager integration components are a separate package, which you must install from the UCMDB interface.

- 1 Log in to your UCMDB system as an administrator.
 -  Log in from the UCMDB file system. The package installation process searches your client's local file system for packages.
- 2 Click the **Settings** tab.
- 3 Click **Package Manager**. UCMDB displays a list of installed packages.
- 4 Click the Deploy Packages to Server (from local disk) button. 
- 5 Click the Add button. 
- 6 Browse to the **integration_packages** folder for your installation. For example:
C:\hp\UCMDB\UCMDBServer\root\lib\integration_packages
- 7 Select the `smintegration.zip` package, and click **Open**.
- 8 Click **OK** to deploy the package.

Add an RMI adapter data store

You must create an RMI data store to act as a source adapter in the UCMDB to Service Manager synchronization. The RMI adapter updates CI relations on the UCMDB system.

- 1 Log in to your UCMDB system as an administrator.
- 2 Click the **Settings** tab.
- 3 Click **Federated CMDB**. UCMDB displays a list of data stores.
- 4 Click the Add button to add a new data store. 
UCMDB displays a Data Store properties window.
- 5 In Adapter, select **CmdbRmiAdapter**.
- 6 In Name, type a unique adapter name. For example, `ucmdb_rmi`.

- 7 In CustomerID, type **1** unless you are using HP Software-as-a-Service, in which case type your customer ID number.
- 8 In Host, type the name of your UCMDB system. For example, **localhost**.

Data Store

Choose the adapter and fill in the connection properties (fields in red are mandatory). Click "Test Connection" to verify that the properties are correct.

Properties

Adapter: CmdbRmiAdapter

Name: ucmdb_rmi

CustomerID: 1

Host: localhost

Port:

User:

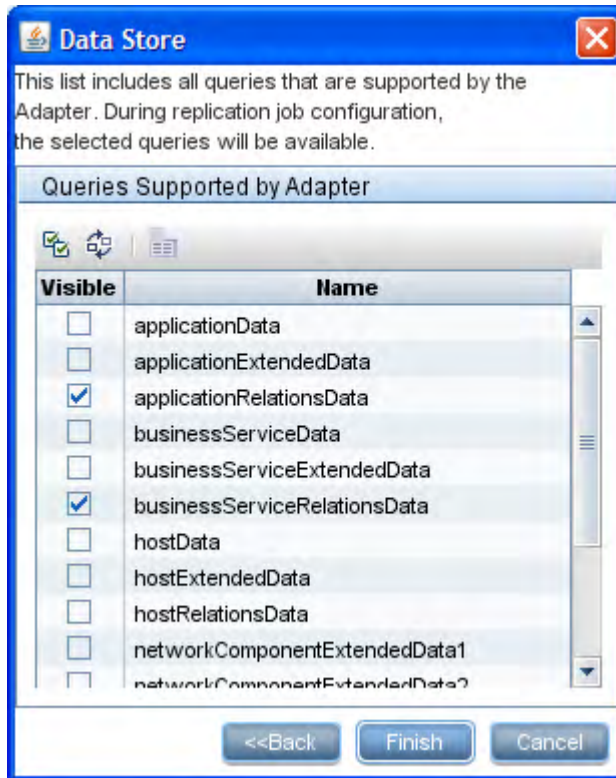
Password:

URL:

Test connection

<=> Back Next >> Cancel


- 9 Click **Next**.
- 10 In Queries Supported by Adapter, select the following queries.
 - applicationRelationsData
 - businessServiceRelationsData



- 11 Click **Finish**. UCMDB creates the data store and displays it in the list.

Add a new Changes adapter data store

You must create a new Changes adapter data store to act as a source adapter in the UCMDB to Service Manager synchronization. The changes adapter updates CI attributes on the UCMDB system.

- 1 Log in to your UCMDB system as an administrator.
- 2 Click the **Settings** tab.
- 3 Click **Federated CMDB**. UCMDB displays a list of data stores.
- 4 Click the Add button to add a new data store. 

UCMDB displays a Data Store properties window.

- 5 In Adapter, select **CmdbChangesAdapter**.
- 6 In Name, type a unique adapter name. For example, **cmdbChanges**.
- 7 In CustomerID, type **1** unless you are using HP Software-as-a-Service, in which case type your customer ID number.
- 8 In Host, type the name of your UCMDB system. For example, **localhost**.

Choose the adapter and fill in the connection properties (fields in red are mandatory). Click "Test Connection" to verify that the properties are correct.

Properties

Adapter: CmdbChangesAdapter

Name: cmdbChanges

CustomerID: 1

Host: localhost

Port:

User:

Password:

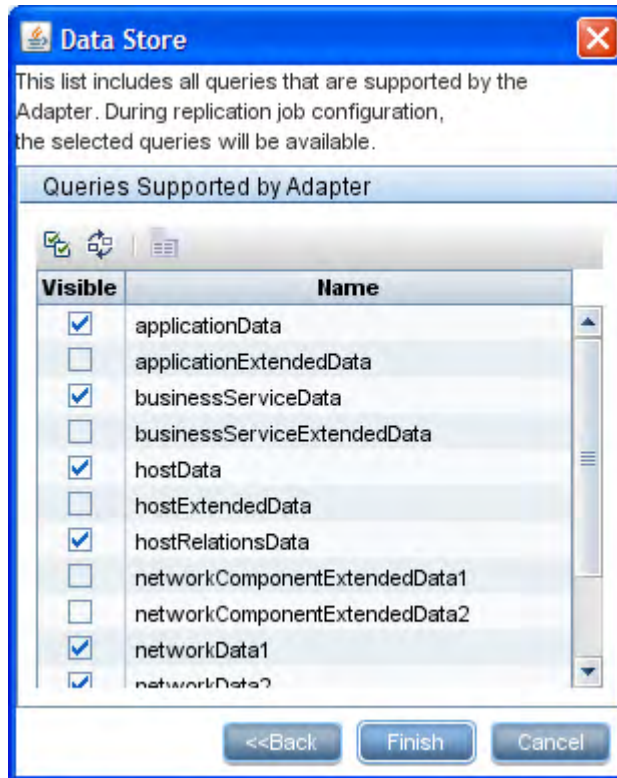
URL:

Test connection

<< Back Next >> Cancel

- 9 Click **Next**.
- 10 In Queries Supported by Adapter, select the following queries.
 - applicationData
 - businessServiceData
 - hostData
 - hostRelationsData


- networkData1
- networkData2
- networkRelationsData
- printerData

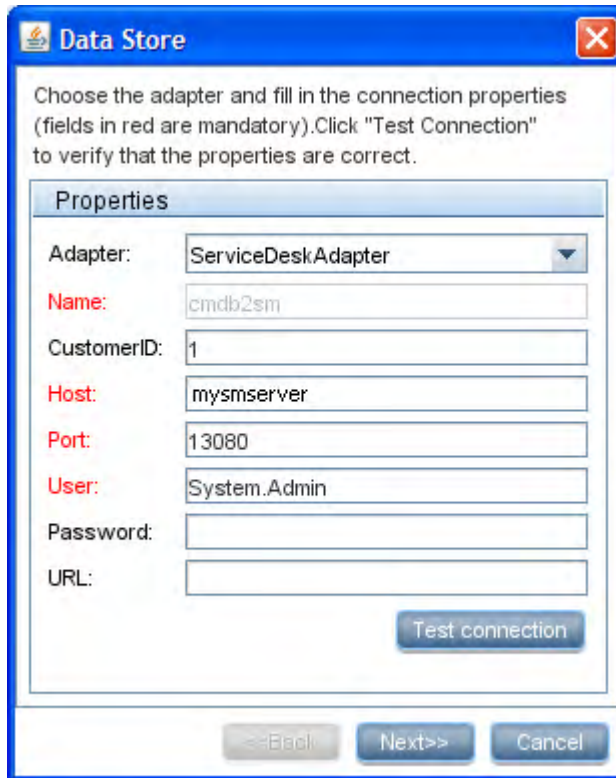


- 11 Click **Finish**. UCMDB creates the data store and displays it in the list.

Add a Service Desk adapter data store

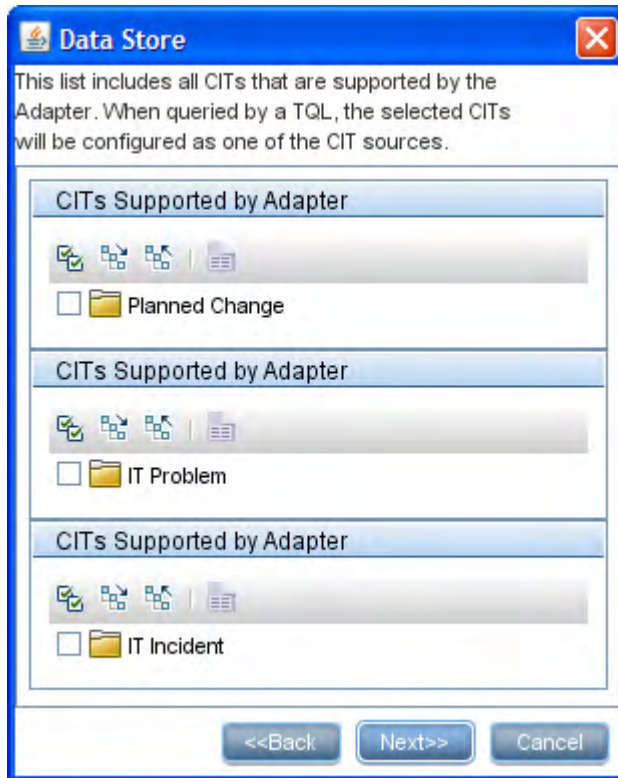
You must create a Service Desk data store to act as a target adapter responsible for sending data from UCMDB to Service Manager. The Service Desk adapter updates CI relations and CI attributes on the Service Manager system.

- 1 Log in to your UCMDB system as an administrator.
- 2 Click the **Settings** tab.
- 3 Click **Federated CMDB**. UCMDB displays a list of data stores.
- 4 Click the **+** button to add a new data store. 
UCMDB displays a Data Store properties window.
- 5 In Adapter, select **ServiceDeskAdapter**.
- 6 In Name, type a unique adapter name. For example, **cmdb2sm**.
- 7 In CustomerID, type **1** unless you are using HP Software-as-a-Service, in which case type your customer ID number.
- 8 In Host, type the name of your Service Manager system. For example, **mysmsserver**.
- 9 In Port, type the communications port of your Service Manager system. For example **13080**.
- 10 In User, type the operator name of a Service Manager system administrator. For example, **System.Admin**.
- 11 For Password, type the password of the Service Manager system administrator. For example, for System.Admin leave the password field blank.

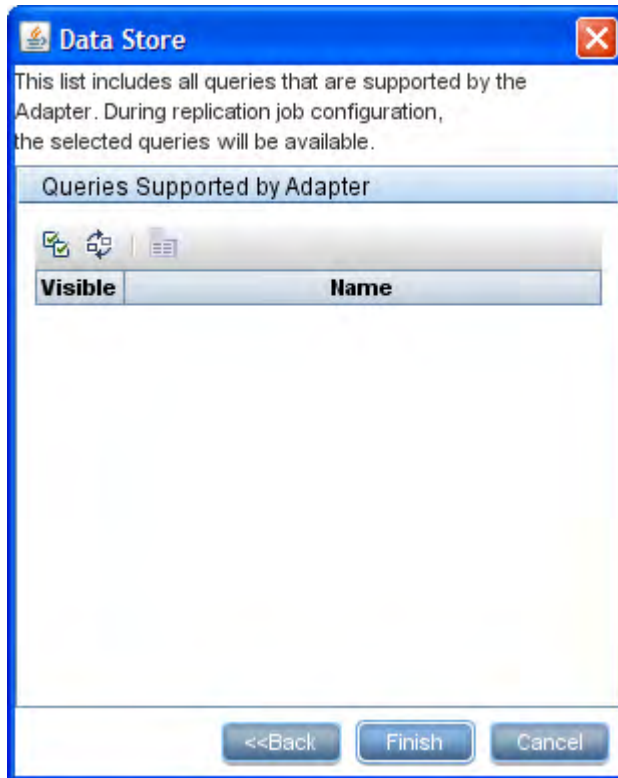


12 Click **Next**.

13 In CI Types supported by adapter, do not select anything, and click **Next**.



- 14 In Queries Supported by Adapter, do not select anything.



- 15 Click **Finish**. UCMDB creates the data store and displays it in the list.

HP Service Manager setup overview

You must complete the following tasks from your Service Manager system to support the integration.

Task 1: Add the UCMDB connection information to the system information record.

See [Add the UCMDB connection information](#) on page 23.

Task 2: If needed, edit the Process Definition record to support a UCMDB 7.52 system.

See [Edit the Process Definition record to support a UCMDB 7.52 system](#) on page 24.

Task 3: Add reconciliation rules to define how the system should determine if incoming CIs from UCMDB match existing Service Manager CI records.

See [Add reconciliation rules](#) on page 25.

Task 4: Add Discovery Event Manager rules to define what the system will do when CI attribute values from UCMDB do not match those in Service Manager.

See [Add Discovery Event Manager rules](#) on page 27.

Add the UCMDB connection information

The integration requires the UCMDB connection information to obtain CI attribute information from the UCMDB system.



Service Manager users are prompted to provide a UCMDB user name and password when they attempt to log in to the UCMDB system from a Service Manager link.

- 1 Log in to your Service Manager system as an administrator.
- 2 Click **System Administration > Base System Configuration > Miscellaneous > System Information Record**.
- 3 Click the **Active Integrations** tab.
- 4 Select the **HP Universal CMDB** option.

The form displays the UCMDB web service URL field.

- 5 In the UCMDB web service URL field, type the URL to the HP Universal CMDB web service API. The URL has the following format:

`http://<UCMDB server name>:<port>/axis2/services/ucmdbSMSService`

Replace *<UCMDB server name>* with the host name of your UCMDB server, and replace *<port>* with the communications port your UCMDB server uses.

- 6 Click **Save**. Service Manager displays the message: Information record updated.
- 7 Log out of the Service Manager system.
- 8 Log back into the Service Manager system with an administrator account. The Actual State tab will be visible in CI records, although you must first complete a replication to populate the Service Manager system with CIs from the UCMDB system.

Edit the Process Definition record to support a UCMDB 7.52 system

If you want the integration to connect to a UCMDB 7.52 system, you must edit a process definition record to update the URL to the UCMDB system. If the integration will connect to a UCMDB 8.0 or greater system, you can skip this step.

- 1 Log in to your Service Manager system as an administrator.
- 2 Click **Tailoring > Document Engine > Processes**.
- 3 In Process Name, type `am.view.ucmdb`, and click **Search**.
- 4 Click the **Initial JavaScript** tab.
- 5 Search for the text “The following line should be uncommented for use with UCMDB 7.5.x.” By default, this is line 3 of the script.
- 6 Remove the comment characters (`//`) from the beginning of the next line, by default, this is line 4.
- 7 Search for the text “The following lines should be uncommented for use with UCMDB 8.0 and above.” By default, these are lines 7 and 8 of the script.
- 8 Add comment characters (`//`) to the beginning of the next two lines, by default, these are lines 7 and 8.
- 9 Click **Save**.

10 Stop and restart your Service Manager system.

Add reconciliation rules

It is possible that your Service Manager system already contains CI records that match CIs in your UCMDB system. Rather than add duplicate CI records to your Service Manager system, you can configure Service Manager to reconcile CI records between the two systems based on the values of specific fields.

Service Manager always attempts to reconcile CI records based on the unique key field of the Service Manager table and the `ucmdb.id` field. You can specify additional fields to reconcile on from the DEM Reconciliation Rules form. If Service Manager finds a matching value in any one of these fields, it updates the Service Manager CI record with the attributes from the incoming UCMDB record.

In order to specify reconciliation fields, you will need to be familiar with the table and field names in both your Service Manager and UCMDB systems. If you want to reconcile on a particular attribute from the UCMDB system, you should verify that there is a corresponding Service Manager managed field for the attribute. Without such a mapping, Service Manager will not know to search for matching values in the CI record.



Not all UCMDB attributes have a corresponding field in Service Manager. You may need to tailor your Service Manager system to add a matching field if one does not already exist.

A reconciliation rule specifies what Service Manager table and field you want to search for matching CI values. It also specifies the sequence in which you want Service Manager to process reconciliation rules. By default, Service Manager processes rules in alphabetical order by field name. For example, Service Manager will reconcile CIs against the `asset.tag` field before reconciling CIs on the `ci.name` field.

To change the order in which Service Manager reconciles CIs, you can add a numeric value to the sequence field. For example, the following reconciliation rules ensure that Service Manager processes CIs by the `ci.name` field prior to reconciling them against the `asset.tag` field.

Table 1 Sample reconciliation rules ordered by sequence

Table Name	Field Name	Sequence
device	ci.name	1
device	asset.tag	2

Create a DEM reconciliation rule

A Discovery Event Manager (DEM) reconciliation rule allows you to specify which Service Manager fields you want to use to determine if an existing CI record matches a CI in a UCMDB system. An administrator typically specifies reconciliation rules prior to starting UCMDB replication jobs so that Service Manager will not create duplicate CI records.

To create a DEM reconciliation rule:

- 1 Log in to your Service Manager system as an administrator.
- 2 Click **Tailoring > Web Services > DEM Reconciliation Rules**. Service Manager displays the DEM Reconcile Record form.
- 3 In **Table Name**, type the name of the Service Manager table containing the field you want to reconcile on.
- 4 In **Field Name**, type the name of the Service Manager field containing the values you want to reconcile on.
- 5 In **Sequence**, type a number to specify what order you want Service Manager to run this rule.
 - ▶ If you do not specify a sequence value, Service Manager will process field names alphabetically.
- 6 Click **New**. Service Manager creates the reconciliation rule.

Add Discovery Event Manager rules

Service Manager uses the Discovery Event Manager (DEM) to define which actions the system should perform when the actual state of an incoming configuration item (CI) record differs from the managed state of a CI record in HP Service Manager. The DEM rules allow you to define whether the Service Manager system adds, updates, or deletes CI records based on incoming UCMDB data.

Service Manager offers the following rules options:

- **Action if matching record does not exist:** This is the action you want Service Manager to take if it cannot find a matching CI record.
 - **Add the record:** (Default) Service Manager will add a CI record when it cannot find a matching record. See [Add reconciliation rules](#) on page 25 to define what fields Service Manager uses to match CI records.
 - **Open an Incident:** Service Manager opens an incident for someone to review the new CI record. The incident enable someone to investigate whether the new CI record is compliant with your business practices.
 - **Open a Change:** Service Manager opens an unplanned change for someone to review the new CI record. The change allows you to investigate whether the new CI record is compliant with your business practices. If the CI record is compliant, the change can be approved. If the CI record is not compliant, then the change can be denied and the CI record removed. The change record lists both the current and proposed attribute values.
- **Action if record exists but unexpected data discovered:** This is the action you want Service Manager to perform if it does not find a matching CI attribute value.
 - **Open a Change:** (Default) Service Manager opens an unplanned change to review the actual state of the CI record. The change allows someone to investigate whether the new attribute value is compliant with your business practices. If the value is compliant, the change can be approved. If the value is not compliant, then the change can be denied and the CI attribute value reverted to its managed state.
 - **Log Results and Update Record:** Service Manager logs the results of the actual state of the CI record, and then update the CI record.

- **Open an Incident:** Service Manager opens an Incident to investigate the actual state of a CI record and determine what actions must be performed or initiated to bring the record into compliance with Service Manager.
- **Action if record is marked for deletion:** This is the action you want Service Manager to perform if an external event specifies that the record needs to be deleted.
 - **Delete record:** (Default) Service Manager automatically deletes the CI record.
 - **Open an Incident:** Service Manager opens an incident to investigate the deleted CI record and determine which actions must be performed or initiated to bring the record into compliance with Service Manager.
 - **Open a Change:** Service Manager opens an unplanned change to review the deleted CI record. The change allows someone to investigate whether the deleted CI record is compliant with your business practices. If CI record is compliant, the change can be approved. If the CI record is not compliant, then the change can be denied and the CI record added back to the system.

Transfer UCMDB CIs to Service Manager

The integration requires a one-time transfer of CIs from UCMDB to Service Manager to populate the Service Manager system with CIs. The integration will then update the list of CIs in Service Manager when UCMDB discovers new CIs or new attribute values. The integration accomplishes the transfer of CI data using replication jobs in the UCMDB system. HP recommends that you schedule these jobs to keep your CIs and CI attributes up to date.

Task 1: Define a Changes replication job between UCMDB and Service Manager.

See [Define a Changes replication job](#) on page 29.

Task 2: Define an RMI replication job between UCMDB and Service Manager.


See [Define an RMI replication job](#) on page 31.

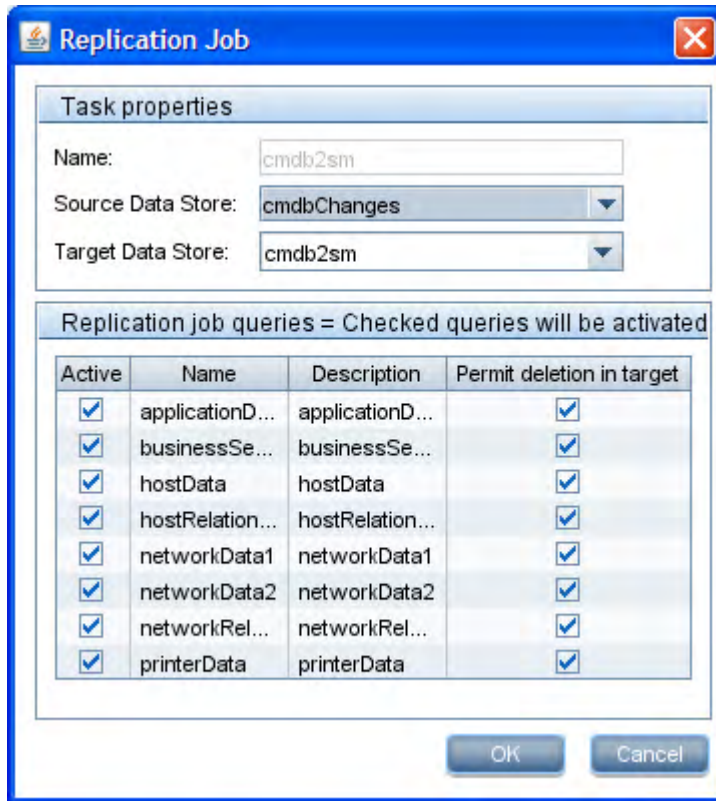
Task 3: Schedule replication jobs to keep CIs and CI attributes up to date.

See [Schedule replication jobs](#) on page 33.


Define a Changes replication job

The changes replication job copies all CIs and CI relations except for business relations from your UCMDB system to your Service Manager system.

- 1 Log in to your UCMDB system as an administrator.
- 2 Go to **Setting - Federated CMDB**.
- 3 Click the **Replication jobs** tab.
- 4 Click **+** to add a new replication job. 
- 5 In Name, type a unique name for the job. For example, **cmdb2sm**.
- 6 In Source Data Store, select the data store name as you created in [Add a new Changes adapter data store](#) on page 16. For example, **CmdbChanges**.
- 7 In Target Data Store, select the data store name as you created in [Add a Service Desk adapter data store](#) on page 19. For example **cmdb2sm**.




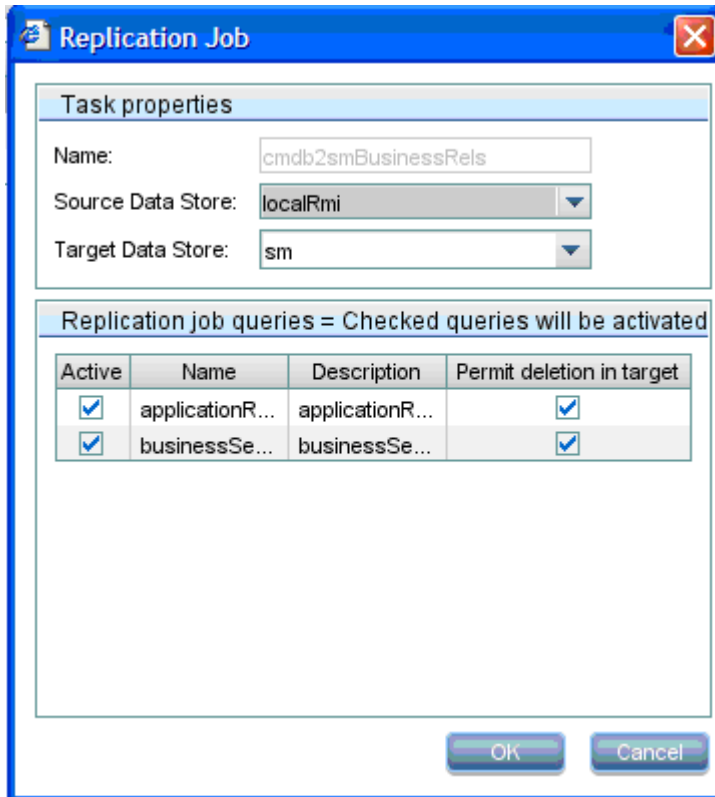
8 Click **OK** to save the replication job.


9 Click the Ad Hoc Full Replication button to run the replication job. 

Define an RMI replication job

The RMI replication job copies all business relations from your UCMDb system to your Service Manager system.

- 1 Log in to your UCMDb system as an administrator.
- 2 Go to **Setting - Federated CMDB**.
- 3 Click the **Replication jobs** tab.
- 4 Click **+** to add a new replication job. 
- 5 In Name, type a unique name for the job. For example, **cmdb2smBusinessReIs**.
- 6 In Source Data Store, select the data store name as you created in [Add an RMI adapter data store](#) on page 14. For example, **localRmi**.
- 7 In Target Data Store, select the data store name as you created in [Add a Service Desk adapter data store](#) on page 19. For example **sm**.





- 8 Click **OK** to save the replication job.
- 9 Click the Ad Hoc Full Replication button to run the replication job. 

Schedule replication jobs

It is a best practice to schedule both the changes and RMI replication jobs to match the discovery schedule of your UCMDB feeders. For example, if your UCMDB feeders send CI data updates on a daily schedule, then the replication jobs should also run on a daily schedule. By using a matching schedule you can ensure that your Service Manager system always has the most current CI data.

To schedule a replication job:

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the **Settings** tab.
- 3 Click **Scheduler**. The Job Scheduler form opens.
- 4 Click **+** to add a new scheduled job. 
- 5 In Name, type a unique name for the job. For example, **changes_schedule**.
- 6 In Actions, click **+** to add a new scheduled action. 
- 7 From the Choose Action list, select **Run Replication Job**, and click **Next**.
- 8 From the Run Replication Job list, select the replication job or jobs you want to schedule. For example, **cmdb2sm** and **cmdb2smBusinessReIs**.
- 9 Click **Finish**. UCMDB displays the action in the action list.
- 10 From the Scheduler section, select the frequency of the action. For example, **Daily**.
- 11 For Time Zone, verify the UCMDB system time zone.
- 12 For Invocation hour, type the hour you want the job to run. For example, 24:00.
- 13 If you only want the scheduled action run during certain specified dates, select the Start at and End by options, and select the dates for which the scheduled job will be valid.
- 14 Click **OK** to create the scheduled job.

3 Tailoring the integration

Integration architecture

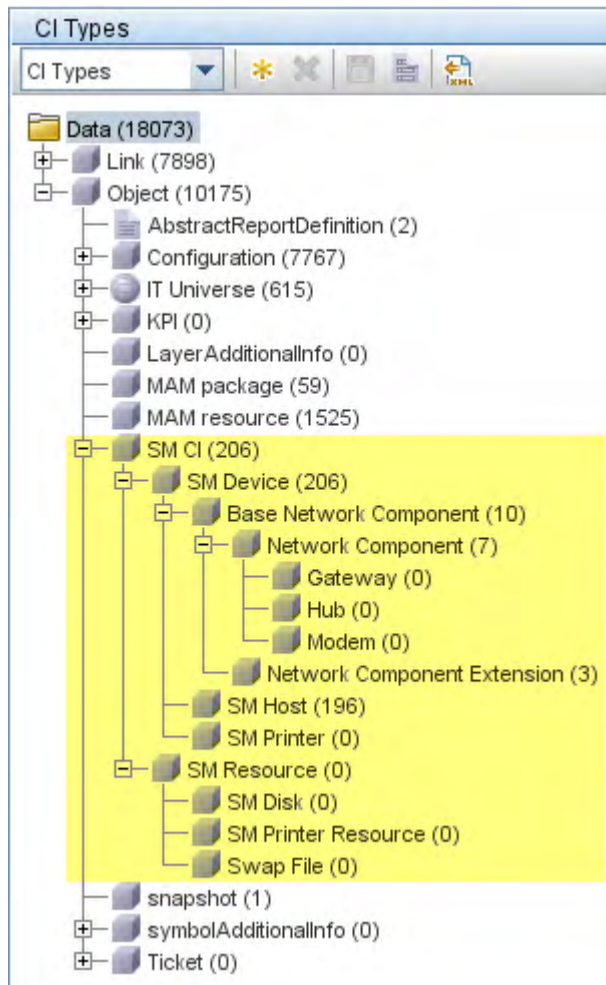
Before you tailor the integration, you should understand how the following components of the out-of-box integration work.

- [Integration class model](#) on page 35
- [Integration TQL queries](#) on page 39
- [Service Manager web services](#) on page 43
- [Service Manager reconciliation rules](#) on page 49
- [Service Manager Discovery Event Manager rules](#) on page 50

Integration class model

The integration comes with a separate class model that maps UCMDB CI types, relationships, and attributes to objects and attributes that Service Manager recognizes. In general, Service Manager recognizes fewer CI types than UCMDB. Service Manager manages certain UCMDB CI types, such as a TCP/IP port, Service Manager as CI attributes rather than a separate CI type. The integration class model groups UCMDB CI types to match how they are managed in Service Manager.

If you want to change the CIs that your UCMDB system sends to your Service Manager system, you will need to edit both the integration class model and the integration queries that support the model. Out-of-box, the integration class model consists of the following CI types and relationships.



In this version of the integration, UCMDB tracks Service Manager CIs separately from the IT Universe. You must change the Object root from the default value of IT Universe in order to see the integration class model.

Out-of-box, UCMDB provides the following CI types for the integration.

Table 2 Integration CI types

CI Type	Description
SM CI	This CI type is a container for all of the CI types and relationships the UCMDB integration provides out-of-box. This CI type inherits attributes from the Object CI type and provides attributes for the SM Device and SM Resource CI types.
SM Device	This CI type is a container for CIs managed as devices in Service Manager. This CI type inherits attributes from the SM CI type and provides attributes for the Base Network Component, SM Host, and SM Printer CI types.
Base Network Component	This CI type is a container for network device CIs. This CI type inherits attributes from the SM Device CI type and provides attributes for the Network Component and Network Component Extension CI types.
Network Component	This CI type is a container for network device CIs. This CI type inherits attributes from the Base Network Component CI type and provides attributes for the Gateway, Hub, and Modem CI types. This CI type is unique to the integration and does not exist in the UCMDB public class model.
Gateway	This is a CI type for network gateways. This CI type inherits attributes from the Network Component CI type. This CI type is unique to the integration and does not exist in the UCMDB public class model.

Table 2 Integration CI types

CI Type	Description
Hub	This is a CI type for network hubs. This CI type inherits attributes from the Network Component CI type. This CI type is unique to the integration and does not exist in the UCMDB public class model.
Modem	This is a CI type for network modems. This CI type inherits attributes from the Network Component CI type. This CI type is unique to the integration and does not exist in the UCMDB public class model.
Network Component Extension	This is a CI type for the Network Component CIs. This CI type inherits attributes from the Base Network Component CI type and also extends attributes from the Host CI type by an SM link relationship.
SM Host	This is a CI type for host CIs. This CI type inherits attributes from the SM Device CI type and also extends attributes from the Host CI type by an SM link relationship.
SM Printer	This CI type manages printer CIs. This CI type inherits attributes from the SM Device CI type and also extends attributes from the Net Printer CI type by an SM link relationship.
SM Resource	This is a container for CIs managed as resources in Service Manager. This CI type inherits attributes from the SM CI type and provides attributes for the SM Printer Resource, SM Disk, and Swap File CI types.

Table 2 Integration CI types

CI Type	Description
SM Disk	This CI type manages hard disk CIs. This CI type inherits attributes from the SM Resource CI type and also extends attributes from the Disk CI type by an SM link relationship.
SM Printer Resource	This CI type manages printer CIs. This CI type inherits attributes from the SM Resource CI type and also extends attributes from the Printer CI type by an SM link relationship.
Swap File	This CI type manages swap file memory CIs. This CI type inherits attributes from the SM Resource CI type and also has a container link relationship to Host CI types. This CI type is unique to the integration and does not exist in the UCMDB public class model.

Integration TQL queries

The integration uses a collection of Topology Query Language (TQL) queries to gather CI attribute information from UCMDB and send it to the Service Manager system. You can view these queries in the **Integration\SM Sync** folder of your UCMDB installation. If you want to change what CI types or attributes are part of the integration, you must also edit the integration queries to support your updated CI types and attributes. Out-of-box, the integration consists of the following queries.

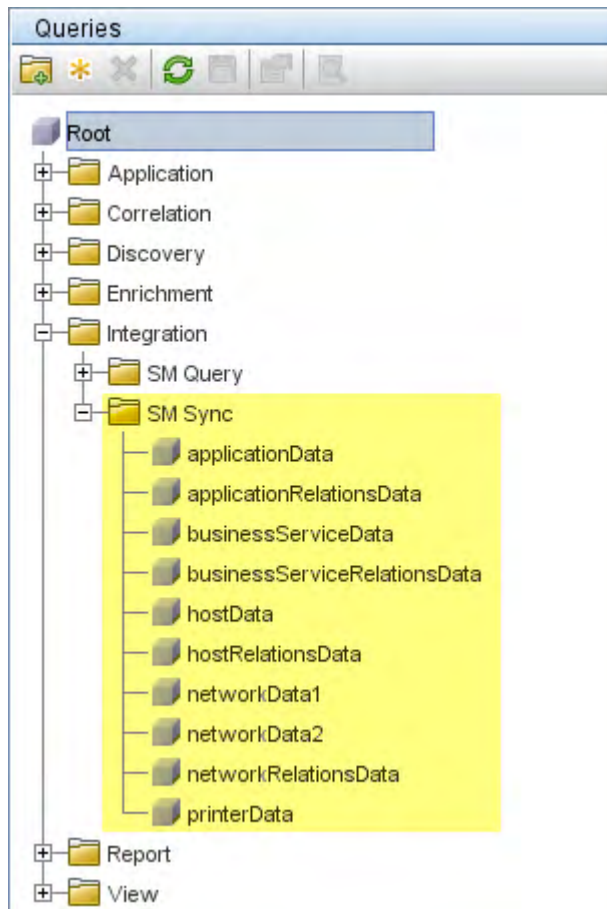


Table 3 Integration queries

Query name	Description
applicationData	This query gathers CI attributes from application CIs.
applicationRelationsData	<p>This query gathers relations between the following components:</p> <ul style="list-style-type: none">• Application and host CIs• Application and network component CIs• Two or more application CIs <p>The query includes compound relations because the relation can extend though a group.</p>
businessServiceData	This query gathers CI attributes from Business Service CIs.
businessServiceRelationsData	<p>This query gathers relations between the following components:</p> <ul style="list-style-type: none">• Business service and application CIs• Business service and host CIs• Business service and network component CIs• Two or more business services <p>The query includes compound relations because the relation can extend through a group.</p>
hostData	<p>This query gathers CI attributes from the following host CI types: host, host_node, lpar, mainframe, nt, terminalserver, unix, and vax.</p> <p>It also gathers related CI attributes from the following CI types through containers and links: IP, Interface, Memory, and SM Host.</p>

Table 3 Integration queries

Query name	Description
hostRelationsData	<p>This query gathers relations between the following components:</p> <ul style="list-style-type: none">• Host and interface CIs• Host and IP CIs• Host and network component CIs• Two or more host CIs <p>The root class of the relation is talk.</p>
networkData1	<p>This query gathers CI attributes from the following host CI types: atmswitch, marconiatmswitch, ras, lb, concentrator, netdevice, firewall, switch, switchrouter, or router.</p> <p>It also gathers related CI attributes from the following CI types through containers and links: Interface, IP, Network Component Extension.</p>
networkData2	<p>This query gathers CI attributes from network component CIs.</p> <p>It also gathers related CI attributes from the following CI types through containers and links: Interface and IP.</p>
networkRelationsData	<p>This query gathers relations between:</p> <ul style="list-style-type: none">• Network component and host CIs• Network component and Interface CIs• Network component and IP CIs• Two or more network component CIs <p>The root class CI type of the relation is talk.</p>
printerData	<p>This query gathers CI attributes from net printer CIs.</p> <p>It also gathers related CI attributes from the following CI types through containers and links: Interface, IP, and SM Printer.</p>

TQL query requirements

The integration requires that any custom TQL queries you create meet certain formatting conditions. Any TQL queries that you want to include in the integration must meet these conditions:

- To query CIs, a query must contain one CI type labeled **Root**. The Root node is the main CI that the UCMDB synchronizes. All other CIs are contained CIs of the Root CI.
- To query relations, a query contain one or more relations labeled **Root**.
- A query must contain only the Root CI and CIs that are directly connected to it. The Root CI is always the top node of the TQL hierarchy.
- A TQL graph cannot have cycles.
- If a query synchronizing relations has cardinality, it must be cardinality **1 . . . ***. Additional cardinality entries must have an OR condition between them.
- The Cmdb Changes adapter does not support compound relation.
- If you only want the integration to synchronize specific CIs, you must first configure the condition on the CIs themselves and then configure the same condition on the TQL query.

Service Manager web services

Service Manager uses web services messages to get and receive CI information from your UCMDB system. Out-of-box, UCMDB sends more CI attribute information than the Service Manager system actually manages. Most of the CI attribute information the UCMDB system provides is for reference only. Service Manager users can view all of the CI attribute information the UCMDB system sends from the Actual State tab of the CI record.

Service Manager publishes several web services for use by the UCMDB integration. The UCMDB system uses the web services to map UCMDB CI types and CI attributes to web services objects the Service Manager system recognizes. If you add UCMDB CI types or CI attributes that you want Service Manager to manage, then you must update one or more of these web services to define them as web services objects. See the Service Manager help for more information about publishing web services.

Managed fields

A Service Manager managed field is a field where the system compares the CI attribute value in the incoming UCMDB web services message to the value in a Service Manager CI record. If the values in the web services message do not match those in the CI record, Service Manager runs a Discovery Event Manager (DEM) rule to determine what action to take.

Service Manager stores the list of managed fields in the **ucmdbIntegration** web service. The ucmdbIntegration web services consists of the following web services objects.

Table 4 Mappings between web service objects, Service Manager tables, and DEM rules

This web service object	Publishes fields from this table	And uses this DEM rule ID
Relationship	cirelationship	ucmdbRelationship
ucmdbApplication	device	ucmdbApplication
ucmdbBusinessService	joinbizservice	ucmdbBusinessService
ucmdbComputer	joincomputer	ucmdbComputer
ucmdbNetwork	joinnetworkcomponents	ucmdbNetwork
ucmdbPrinter	joinofficeelectronics	ucmdbPrinter

The DEM rule determines which of the fields that are published as web services objects are fields managed by the integration. Only value changes in managed fields trigger the DEM rule.

The following sections list the fields published as web services objects and whether they are managed fields in an out-of-box Service Manager system. You can use this reference to determine if you need to publish a field as a web service object, and also if you need to create a DEM rule for the object.

Object Name: Relationship

Service Manager manages the following fields from the cirelationship table:

Table 5 Web service and managed fields of the Relationship object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
relationship.name	RelationshipName	
logical.name	ParentCI	
related.cis	ChildCIs	Yes
relationship.type	RelationshipType	
relationship.subtype	RelationshipSubtype	

Object Name: ucmdbApplication

Service Manager publishes the following fields from the device table:

Table 6 Web service and managed fields of the ucmdbApplication object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
ucmdb.id	UCMDBId	
ci.name	ApplicationName	Yes
subtype	Subtype	

Object Name: ucmdbBusinessService

Service Manager publishes the following fields from the joinbizservice table:

Table 7 Web service and managed fields of the ucmdbBusinessService object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
ucmdb.id	UCMDBId	
ci.name	ServiceName	Yes
vendor	ServiceProvider	Yes
subtype	Subtype	Yes

Object Name: ucmdbComputer

Service Manager manages the following fields from the joincomputer table:

Table 8 Web service and managed fields of the ucmdbComputer object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
ucmdb.id	UCMDBId	
default.gateway	DefaultGateway	Yes
operating.system	OS	Yes
os.version	OSVersion	Yes
bios.id	BIOSId	Yes
network.name	DNSName	Yes
addlIPAddr[addlIPAddress]	AddlIPAddress	Yes
addlIPAddr[addlSubnet]	AddlSubnet	Yes
addlMacAddress	AddlMacAddress	Yes

Table 8 Web service and managed fields of the ucldbComputer object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
physical.mem.total	PhysicalMemory	Yes
subtype	Subtype	Yes
building	Building	Yes
room	Room	Yes
floor	Floor	Yes
machine.name	HostName	Yes

Object Name: ucldbNetwork

Service Manager manages the following fields from the joinnetworkcomponents table:

Table 9 Web service and managed fields of the ucldbNetwork object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
ucldb.id	UCMDBId	
default.gateway	DefaultGateway	Yes
operating.system	OS	Yes
os.version	OSVersion	Yes
network.name	DNSName	Yes
addlIPAddr[addlIPAddress]	AddlIPAddress	Yes
addlIPAddr[addlSubnet]	AddlSubnet	Yes
addlMacAddress	AddlMacAddress	Yes

Table 9 Web service and managed fields of the ucldbNetwork object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
subtype	Subtype	Yes
building	Building	Yes
room	Room	Yes
floor	Floor	Yes

Object Name: ucldbPrinter

Service Manager manages the following fields from the joinofficeelectronics table:

Table 10 Web service and managed fields of the ucldbPrinter object

Field published as web service object	Caption used in web service messages	Is the field a managed field?
ucldb.id	UCMDBId	
default.gateway	DefaultGateway	Yes
network.name	DNSName	Yes
addlIPAddr[addlIPAddress]	AddlIPAddress	Yes
addlIPAddr[addlSubnet]	AddlSubnet	Yes
addlMacAddress	AddlMacAddress	Yes
subtype	Subtype	Yes
building	Building	Yes
room	Room	Yes
floor	Floor	Yes

Service Manager reconciliation rules

Service Manager reconciliation rules allow the integration to identify CI records in your Service Manager system that match CIs in your UCMDB system. Service Manager attempts to reconcile CI records with every push of CI attributes from your UCMDB system. The integration uses the following workflow to match UCMDB CIs with Service Manager CIs.

- 1 The UCMDB system sends a web service message to Service Manager containing the latest CI attribute data.
- 2 Service Manager scans the web service message for the CI `ucmdb.id` value.
 - ▶ Out-of-box, Service Manager does not display the `ucmdb.id` field on CI record forms to prevent users from changing the value. If you want to add this value to your forms, you can find the `ucmdb.id` field defined in the device table. HP recommends that you make this a read-only field.
- 3 Service Manager searches for an existing CI record that has the same `ucmdb.id` value.
- 4 If Service Manager finds a CI with a matching `ucmdb.id` value, no reconciliation is needed. Service Manager compares the UCMDB CI attributes to the Service Manager managed fields and runs the appropriate Discovery Event Manager (DEM) rules as needed.
- 5 If Service Manager cannot find a CI with a matching `ucmdb.id` value, it runs the reconciliation rules.
- 6 Service Manager searches for an existing CI record with the same reconciliation field values.
- 7 If Service Manager finds a CI with a matching reconciliation field value, it updates the CI record with the `ucmdb.id` value of matching UCMDB CI. Service Manager compares the UCMDB CI attributes to the Service Manager managed fields and runs the appropriate DEM rule as needed.
- 8 If Service Manager cannot find a CI with a matching reconciliation field value, it runs the DEM rule for “Action if matching record does not exist.” Out-of-box, the DEM rule has Service Manager create a new CI record. Service Manager creates the CI record using the `ucmdb.id` value of incoming UCMDB CI.

Performance implications

Because Service Manager attempts to reconcile CIs with every push, the number of reconciliation fields you have will affect the integration's performance. The more reconciliation rules you have, the more searches Service Manager must perform to match CIs. To improve the performance of reconciliation searches, you should choose reconciliation fields that are unique keys of the underlying Service Manager table. For example, if you want to reconcile CI records in the device table, use the `logical.name` field as a reconciliation field because it is a unique key. See [Add reconciliation rules](#) on page 25 to create a reconciliation rule.

Dependence on DEM rules

Service Manager uses the “Action if matching record does not exist” DEM rule whenever it cannot reconcile CIs. You should review the DEM settings and decide if they meet your business standards prior to the initial push of CIs from UCMDB to Service Manager. For example, you can have Service Manager create a change request for every CI in the initial CI push by selecting the “Open a change” option.

Service Manager Discovery Event Manager rules

You only need to create Discovery Event Manager (DEM) rules if you want to accomplish any of the following custom actions:

- [Change the conditions under which a DEM rule runs](#) on page 50
- [Change the action the DEM rule takes](#) on page 52
- [Update the list of managed fields for a CI type](#) on page 52
- [Create custom JavaScript to open change or incident records](#) on page 52

Change the conditions under which a DEM rule runs

Service Manager will only run a DEM rule if the condition field evaluates to true. Out-of-box, only the `ucmdbApplication` DEM rule has a condition statement that restricts when the rule runs. All the other integration DEM rules will always run by default.

You can update a DEM rule's condition statements if you want to restrict when Service Manager runs your DEM rules. For example, adding the following condition to the ucldbComputer DEM rule restricts the rule to desktop CIs.

```
subtype in $L.file="Desktop"
```

You can also use the condition field to create multiple DEM rules that apply to the same table name. For example, the following DM rules both apply to the joincomputer table.

Table 11 DEM rules using different conditions to affect the same table

DEM rule Id	Table Name	Condition
ucldbComputer	joincomputer	subtype in \$L.file!="Desktop"
ucldbDesktop	joincomputer	subtype in \$L.file="Desktop"

Typically, you will only need to add conditions if your business processes require the integration to take different actions with certain CI types or SLAs.

Change the action the DEM rule takes

Out-of-box, the integration DEM rules take the following actions:

- Add a CI record when the UCMDB data does not match an existing Service Manager CI record
- Open a Change when the UCMDB CI attribute data does not match the CI attribute data in the Service Manager CI record
- Delete a CI record when the UCMDB data specifies that the CI has been deleted

You can change the integration DEM rules to meet your business processes. For example, you could use the `ucmdbComputer` DEM rule to open a change when the integration finds a non-desktop CI with unexpected data, and use the `ucmdbDesktop` DEM rule to log results and update the record when the integration finds a desktop CI with unexpected data.



If you want to use the Change Management verification and Change Management validation features of the integration, then your DEM rules must use the Open a Change option for the “Action if record exists but unexpected data discovered” event.

Update the list of managed fields for a CI type

If you add CI attributes to your UCMDB system that you want to include in the integration, you must also create matching managed fields in Service Manager. Each managed field must have a corresponding web services object definition in order to receive CI attribute updates from your UCMDB system. See [Add a CI attribute to the integration](#) on page 59 and [Add a CI Type to the integration](#) on page 76 for information on how to add managed fields.

Create custom JavaScript to open change or incident records

Service Manager uses the `discoveryEvent` JavaScript to create CI names and to set the values of required fields when opening change or incident records. Out-of-box, the script uses the following default values.

Default values to create a new CI

You can update the createCIName and populateNewCI functions to set the following CI values.

Table 12 Default values used to create a new CI

CI attribute	Default value defined in discoveryEvent
record.logical_name	System generated ID number
record.assignment	AUTO
record.istatus	Installed
record.os_name	Value in record.operating_system

Default values to create a new change

You can update the populateChange function to set the following change values.

Table 13 Default values used to create a new change

CI attribute	Default value defined in discoveryEvent
change.category	Unplanned Change
change.reason	Value in reason
change.initial_impact	3
change.severity	3
change.coordinator	Change.Coordinator
change.requested_by	discovery
change.status	initial

Default values to create a new incident

You can update the populateIncident function to set the following incident values.

Table 14 Default values used to create a new incident

CI attribute	Default value defined in discoveryEvent
incident.category	incident
incident.subcategory	hardware
incident.product_type	missing or stolen
incident.assignment	Hardware
incident.initial_impact	3
incident.severity	3
incident.logical.name	Value of id
incident.site_category	C
incident.contact_name	ANALYST, INCIDENT
incident.affected_item	MyDevices

Making the integration components visible in UCMDB

In an out-of-box UCMDB installation, the integration components are not visible after setup because they are located higher up in the class model than the default view displays. While you do not have to make the integration components visible to use the integration, you do have to make them visible if you want to tailor the integration. You must complete the following tasks to make the integration components visible in the UCMDB interface.

Task 1: [Change the Link Root setting to link.](#)

See [Change the Link Root setting](#) on page 55.




Task 2: [Change the Object Root to data.](#)


See [Change the Object Root setting](#) on page 57.

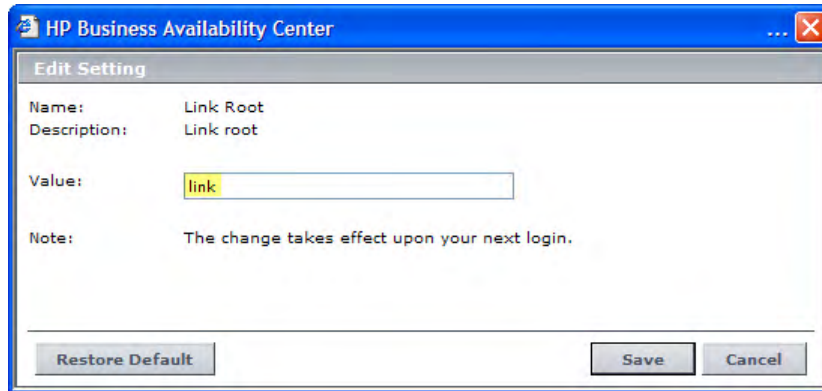
Change the Link Root setting

In an out-of-box UCMDB installation, the Link Root value is `it_world_links`. All of the integration links are in the SM Link node, which is not visible from the `it_world_links` node. If you want to view or change the integration links, you must change the value of the Link Root setting to a node higher up in the class model.

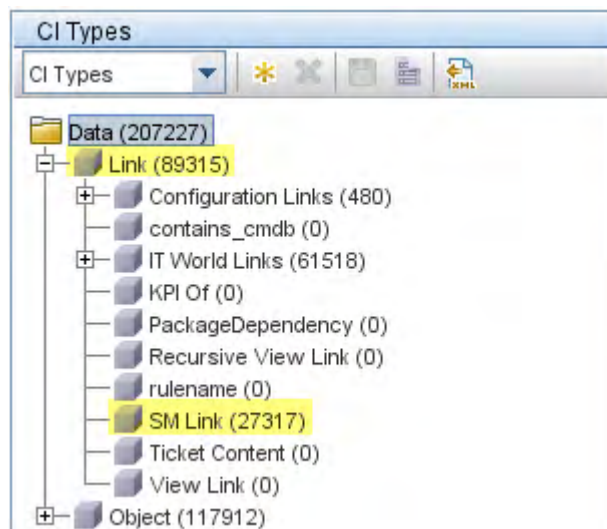
- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the **Settings** tab.
- 3 Click **Infrastructure Settings Manager**.
- 4 For Select Context, select the **All** option.
- 5 Scroll down to the UCMDB Common - General Settings section.

UCMDB common - General Settings		
Name ▲	Description	Value
Candidates For Deletion Refresh Time	Candidates for deletion refresh time	86400000 
Link Root	Link root	it_world_links 
Object Root	Object root	it_world 

- 6 Click the edit button for Link Root. 
- 7 Change the Link Root Value to **link**.






- 8 Click **Save**.
- 9 Logout and log back in to UCMDB.
- 10 To view the integration links, open the CI Types manager and browse to **Data > Link > SM Link**.




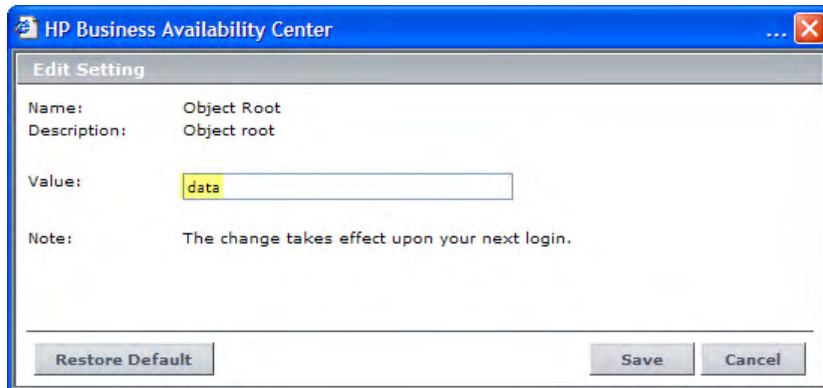
Change the Object Root setting

In an out-of-box UCMDB installation, the Object Root value is `it_world`. All of the integration objects are in the SM CI node which is not visible from the `it_world` node. If you want to view or change the integration objects, you must change the value of the Object Root setting to a node higher up in the class model.

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the **Settings** tab.
- 3 Click **Infrastructure Settings Manager**.
- 4 For Select Context, select the **All** option.
- 5 Scroll down to the UCMDB Common - General Settings section.

UCMDB common - General Settings		
Name ^	Description	Value
Candidates For Deletion Refresh Time	Candidates for deletion refresh time	86400000 
Link Root	Link root	it_world_links 
Object Root	Object root	it_world 

- 6 Click the edit button for Object Root. 
- 7 Change the Object Root Value to **data**.



HP Business Availability Center

Edit Setting

Name: Object Root
Description: Object root

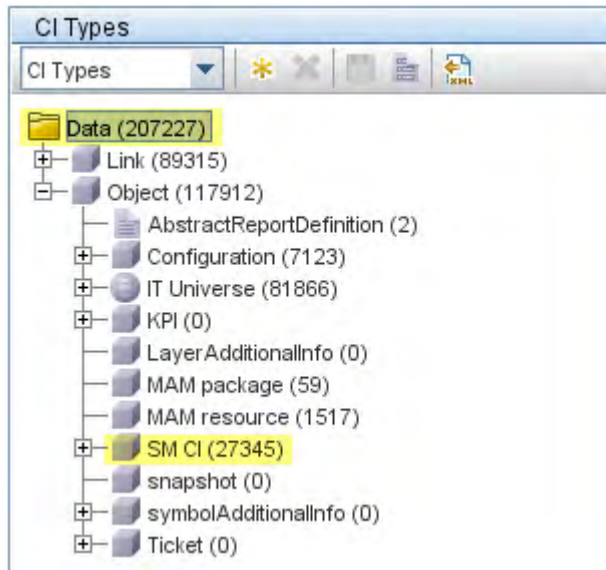
Value:

Note: The change takes effect upon your next login.

Restore Default Save Cancel

- 8 Click **Save**.

- 9 Log out, and then log in again to UCMDB.
- 10 To view the integration CI types, open the CI Types manager and browse to **Data > Object > SM CI**.



Tailoring options

The integration offers the following tailoring options:

- [Add a CI attribute to the integration](#) on page 59
- [Add a CI Type to the integration](#) on page 76

Add a CI attribute to the integration

You can use the following steps to add a CI attribute to the integration.

Task 1: [Does the CI attribute already exist in the UCMDB class model?](#)

Yes. Go to [Task 3](#).

No. Go to [Task 2](#).

Task 2: [Add the CI attribute to the UCMDB class model.](#)

See [Add a CI attribute to the UCMDB class model](#) on page 60.

Task 3: [Enable calculation for the CI attribute.](#)

See [Enable calculation for the CI attribute](#) on page 62.

Task 4: [Map the CI attribute to a web service object](#)

See [Map the CI attribute to a web service object](#) on page 66.

Task 5: [Reload the code base for the ServiceDeskAdapter.](#)

See [Reload the code base for the ServiceDeskAdapter](#) on page 69.

Task 6: [Create a web service object to support the CI attribute.](#)

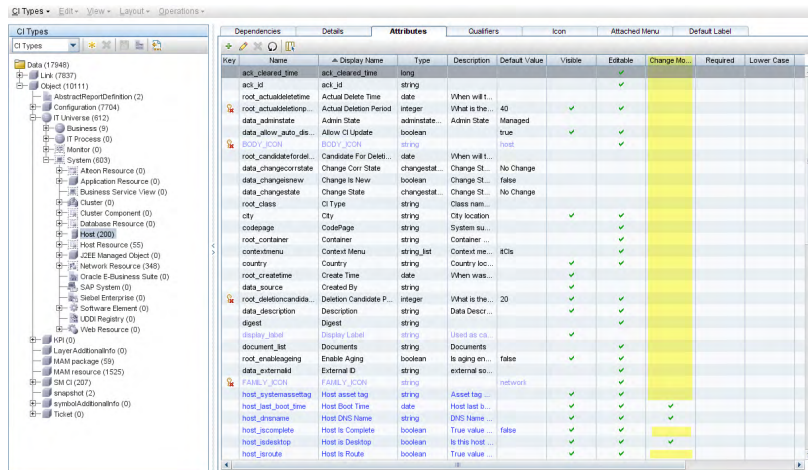
See [Create a web service object to support the CI attribute](#) on page 71.

Task 7: [Add a managed field to monitor changes in the CI attribute.](#)

See [Add a managed field to support the CI attribute](#) on page 72.

Add a CI attribute to the UCMDB class model

The integration class model only uses a subset of the CI attributes available from your UCMDB system. Out-of-box, the integration consists of CI attributes that are typically managed from a Service Manager system such as host name and host DNS name. Before creating a new UCMDB CI attribute, you should determine if there are any existing CI attributes in your UCMDB system that provide the discovery data you want. In most cases, there is an existing attribute tracking the discovery data you want to add to the integration. For example, if you review the attributes of the Host CI, you see that there are many attributes available to be added to the integration (integration attributes must have the Change Monitored setting enabled).



The screenshot shows the UCMDB CI Types interface. On the left is a tree view of CI types, with 'Host (200)' selected. The main pane displays a table of attributes for the Host CI type. The table has columns for Key, Name, Display Name, Type, Description, Default Value, Visible, Editable, Change Mo., Required, and Lower Case. The 'Change Mo.' column is highlighted in yellow, indicating that these attributes are change-monitored.

Key	Name	Display Name	Type	Description	Default Value	Visible	Editable	Change Mo.	Required	Lower Case
act_clear_time	act_clear_time	act_clear_time	long							
act_24	act_24	act_24	string							
root_actualexpiretime	Actual Delete Time	Actual Delete Time	date	When will t...						
root_actualexpireperiod	Actual Delete Period	Actual Delete Period	integer	When is the...	40					
data_adminstate	Admin State	Admin State	adminstate	Admin State	Managed					
data_allow_auto_dis	Allow CI Update	Allow CI Update	boolean	true						
BSOY_ICON	BSOY_ICON	BSOY_ICON	string	icon						
root_candidatefordel	Candidate For Deleti...	Candidate For Deleti...	date	When will t...						
data_changeconstate	Change Con State	changeconstat...	changeconstat...	Change St...	No Change					
data_changewview	Change In View	Change In View	boolean	Change St...	false					
data_changestate	Change State	changeconstat...	changeconstat...	Change St...	No Change					
root_class	CI Type	CI Type	string	Class nam...						
city	City	City	string	City location						
collection	Collection	Collection	string	System na...						
root_container	Container	Container	string	Container...						
contentmenu	Content Menu	Content Menu	string_list	Content me...	ACis					
country	Country	Country	string	Country loc...						
root_createdtime	Create Time	Create Time	date	When was...						
data_source	Created By	Created By	string							
root_deletioncandida	Deletion Candidate P...	Deletion Candidate P...	integer	What is the...	20					
data_description	Description	Description	string	Data Descr...						
digit	Digit	Digit	string							
display_label	Display Label	Display Label	string	Used as co...						
document_list	Documents	Documents	string	Documents						
root_enableaging	Enable Aging	Enable Aging	boolean	Is aging en...	false					
data_externalid	External ID	External ID	string	external id...						
FAMILY_ICON	FAMILY_ICON	FAMILY_ICON	string	network						
host_systemsetting	Host asset tag	Host asset tag	string	Asset tag ...						
host_last_job_time	Host Job Time	Host Job Time	date	Host last t...						
host_hostname	Host DNS Name	Host DNS Name	string	DNS Name ...						
host_iscomplete	Host is Complete	Host is Complete	boolean	True value	false					
host_isdesktop	Host is Desktop	Host is Desktop	boolean	Is this host...						
host_isroute	Host is Route	Host is Route	boolean	True value						


The following steps illustrate how to add a new CI attribute to an existing CI type. This scenario is not the expected typical case. Typically, you would add an existing CI attribute to the integration.



The integration does not require any special steps to add a CI attribute to the UCMDB class model. You can use the standard CI attribute creation procedures to add a CI attribute. For more information on CI attribute creation, see the *HP Universal CMDB CI Attribute Customization Guide*.

To add a CI attribute to the UCMDB class model:

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the **Modeling** tab.
- 3 Click **CI Type Manager**.
- 4 Select the CI type to which you want to add a new CI attribute from the CI Types navigation tree. For example, **Data > Object > IT Universe > System > Software Element > Database**.
- 5 Click the **Attributes** tab.


- 6 Click the add button. 

The Add Attribute window opens.

- 7 In Attribute Name, type the unique name you want to use for the new CI attribute. For example, **database_owner**.



The name cannot include any of the following characters: ` / \ [] : | < > + = ; , ? *.

- 8 In Display Name, type the name you want UCMDB to display in the interface. For example, **Database Owner**.
- 9 In Description, type a description of the new CI attribute. This is an optional field. For example, **System user who owns the database**.
- 10 In Attribute Type, select either Primitive or Enumeration/List. For example, select **Primitive** and select **string**.
- 11 In Value Size, type the maximum character length the attribute can have. For example, **300**.
- 12 In Default Value, type the value to be used when no other value is available. For example, leave the default value blank.
- 13 In the Advanced section, enable the **Change Monitored** option.
- 14 Click **OK** to save the attribute.
- 15 Click the save button to save attribute changes to the CI type. 

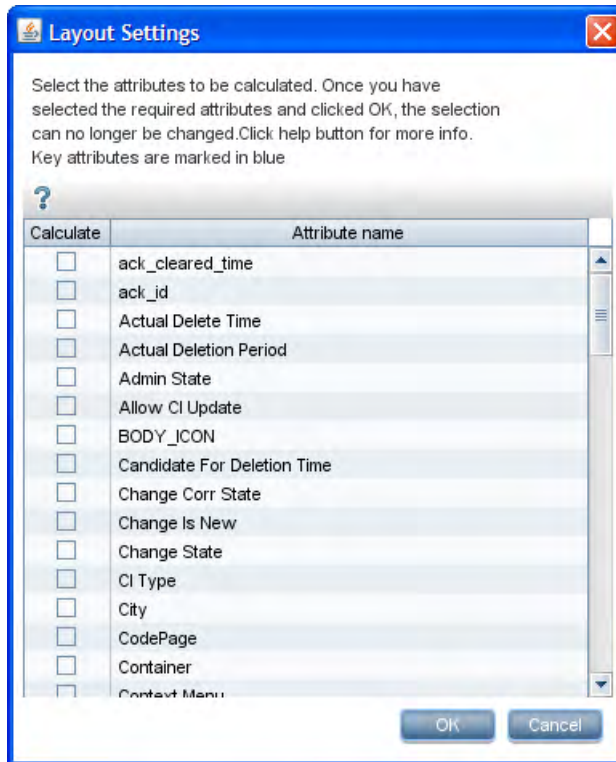
Enable calculation for the CI attribute

To add a CI attribute to the integration, you must enable the calculation layout setting from the TQL query that synchronizes the CI type. You must know what CI type contains the CI attributes you want to add to the integration.

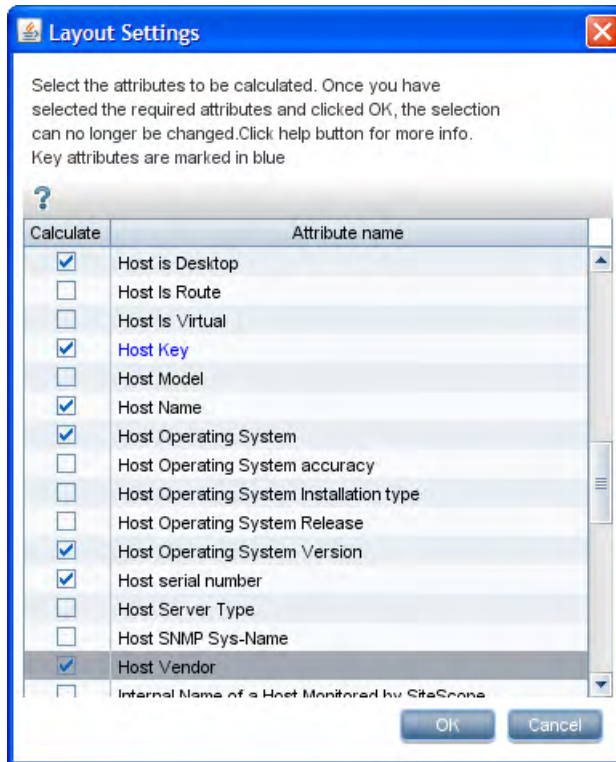



Keep a list of the attributes you enable, because you will need to create a matching XSL transformation for each attribute you enable.

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the Modeling tab.
- 3 Click **Query Manager**.
- 4 From the Queries navigation tree, click **Integration > SM Sync**.
- 5 Select the query that manages the CI type whose attributes you want to add to the integration. For example, **hostData**. UCMDB displays the TQL for the integration query.
- 6 Select the node from the TQL layout that contains the CI attribute you want to add to the integration. For example, **Root**.
- 7 Click the **Edit** button from the Information Pane. The Node properties window opens.
- 8 Click **Advanced layout settings**. The Layout Settings window opens.



- 9 Enable the Calculate option for the CI attribute you want to include in the integration. For example, Host vendor.




- 10 Click **OK** to close the advanced layout settings window.
- 11 Click **OK** to save the node properties.
- 12 Click the save button to save the TQL query. 

Enable the Change Monitored option for CI attributes

In order for the integration to send updates of CI attribute values, you must enable the Change Monitored option for each CI attribute you want to be part of the integration.

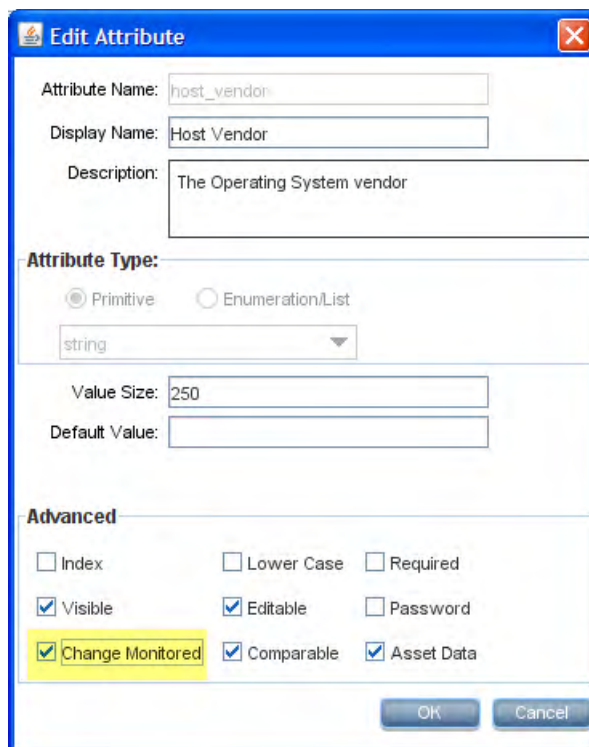
- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the Modeling tab.
- 3 Click **CI Type Manager**.

- 4 Select the CI Type containing the attributes you want to edit from the navigation tree. For example, **Data > Object > It Universe > System > Host**.
- 5 Click the **Attributes** tab. UCMDB displays a list of the attributes associated with this CI type.
- 6 Select an attribute you want to be part of the integration. For example, Host Vendor.

- 7 Click the edit button. 

The Edit Attribute window opens.

- 8 From the Advanced section, enable the **Change Monitored** option.



Edit Attribute

Attribute Name:

Display Name:

Description:

Attribute Type:

Primitive Enumeration/List

Value Size:

Default Value:


Advanced

Index Lower Case Required

Visible Editable Password

Change Monitored Comparable Asset Data

- 9 Click **OK** to accept the changes.

- 10 Click the save button to save the TQL query. 

Map the CI attribute to a web service object

The integration uses the Service Desk Adapter to transform UCMDB CI attributes to web services objects recognized by Service Manager. The Service Desk Adapter uses XSL transformation files to convert UCMDB TQL queries into a properly formatted Service Manager web services messages.

Out-of-box, each integration query has a corresponding XSL transformation file that maps to a particular CI type in UCMDB. In addition, each attribute for which you enabled calculation requires its own entry in the XSL transformation file. Without an XSL transformation entry, Service Manager cannot receive any CI attribute updates from your UCMDB system.

If you want to add a new attribute to the integration, you must edit the XSL transformation file for the parent CI type and add an entry for the CI attribute. See [Integration TQL queries](#) on page 39 for information about which CI types each query manages. In order to create a proper XSL mapping, you must be familiar with the service and object names Service Manager publishes as Web services. See the Service Manager help for information on publishing tables and columns as Web service objects.

The following steps illustrate how to map a UCMDB CI attribute called `host_vendor` to a Service Manager web service object called `OSVendor`.

To map a CI attribute to a web service object:

- 1 Log in to the UCMDB file system with an administrator account.
- 2 Browse to the Service Desk Adapter path in the file system. For example:
**C:\hp\UCMDB\UCMDBServer\j2f\fcmdb\CodeBase
\ServiceDeskAdapter**
- 3 Use a text editor to open the XSL transformation file that manages parent CI type of your CI attribute. For example, open `host_data.xslt` to add an attribute to the `hostData` TQL query.
- 4 Find the element that defines the Service Manager table name where the integration will store CI attribute values. For example, the element `<file.device>` will store CI attributes in the Service Manager `device` table.
- 5 Within the table naming element (`<file.device>`), you will see an element of the following format that defines how to transform each UCMDB CI attribute into a web service object:

```
<xsl:for-each select="@CI_attribute_name">
```

```
    <ObjectName><xsl:value-of select="." /></ObjectName>
  </xsl:for-each>
```

@CI_attribute_name is the name of attribute in the UCMDB system.

ObjectName is the name of a web service object published by the Service Manager system.

Figure 1 CI attributes in the host_data.xslt XSL transformation file

```
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="/host">
- <model>
  <keys />
- <instance>
- <file.device>
  <!-- import:cmdb_root_attributes.xslt -->
- <xsl:choose>
- <xsl:when test="@host_isdesktop='true'">
  <Subtype>Desktop</Subtype>
</xsl:when>
- <xsl:otherwise>
- <xsl:for-each select="@friendlyType">
  - <Subtype>
    <xsl:value-of select="." />
  </Subtype>
</xsl:for-each>
</xsl:otherwise>
</xsl:choose>
- <xsl:for-each select="//sm_host">
  <!-- import:sm_device_attributes.xslt -->
</xsl:for-each>
- <xsl:for-each select="@host_os">
- <OS>
  <xsl:value-of select="." />
</OS>
</xsl:for-each>
- <xsl:for-each select="@host_dnsname">
- <DNSName>
  <xsl:value-of select="." />
</DNSName>
</xsl:for-each>
</file.device>
```

Figure 2 Matching CI attributes in the ucmdbComputer web service

External Access Definition

Service Name: Released

Name: Deprecated

Object Name:

Allowed Actions Expressions **Fields**

Field	Caption	Type
ucmdb.id	UCMDBId	
default.gateway	DefaultGateway	
operating.system	OS	
os.version	OSVersion	
bios.id	BIOSId	
network.name	DNSName	
addIPAddr[addIPAddress]	AddIPAddress	
addIPAddr[addSubnet]	AddSubnet	
addMacAddress	AddMacAddress	
physical.mem.total	PhysicalMemory	
subtype	Subtype	
building	Building	
room	Room	
floor	Floor	
machine.name	HostName	

- 6 Copy an existing XSL transformation element to use it as a template to create a new transformation entry.
- 7 Paste the new XSL transformation element within the proper table naming element. For example, `<file.computer>`.
- 8 Update the CI attribute name and web service object name within the new element to match the attribute you want to add to the integration. For example, create the following XSL transformation element to add the `host_vendor` CI attribute to the integration.

```
<xsl:for-each select="@host_vendor">
  <OSVendor><xsl:value-of select="." /></OSVendor>
</xsl:for-each>
```

Figure 3 New attribute in the `host_data.xslt` XSL transformation file

```
- <file:computer>
- <xsl:for-each select="@host_hostname">
  - <HostName>
    <xsl:value-of select="." />
  </HostName>
</xsl:for-each>
- <xsl:for-each select="@host_osversion">
  - <OSVersion>
    <xsl:value-of select="." />
  </OSVersion>
</xsl:for-each>
- <xsl:for-each select="@host_vendor">
  - <OSVendor>
    <xsl:value-of select="." />
  </OSVendor>
</xsl:for-each>
- <xsl:for-each select="@host_serialnumber">
  - <BIOSId>
    <xsl:value-of select="." />
  </BIOSId>
</xsl:for-each>
```

- 9 Save the XSL transformation file.

Reload the code base for the ServiceDeskAdapter

In order for the integration to use your custom CI attributes, you must reload the code base of the ServiceDeskAdapter.

To reload the code base for the ServiceDeskAdapter:

- 1 Open a web browser and browse to the following URL:


`http://<UCMDB server>:8080/jmx-console/HtmlAdaptor?action=inspectMBean&name=Topaz%3AService%3DFCmdb+Config+Services`

UCMDB server is the fully qualified domain name for your UCMDB system.

- 2 When prompted, provide your UCMDB administrator user name and password.

The JMX MBean View web page opens.

Figure 4 JMX MBean View



JMX MBean View

MBean Name: **Domain Name:** Topaz
service: FCmdb Config Services

MBean Java Class: com.mercury.topaz.cmdb.server.fcmdb.base.jmx.config.ConfigJmxServices

[Back to Agent View](#) [Refresh MBean View](#)


- 3 Scroll to the section for the loadOrReloadCodebaseForAdapterId() MBean.
- 4 For customerID, type a ParamValue of **1**.
- 5 For adapterId, type a ParamValue of **ServiceDeskAdapter**.

java.lang.String loadOrReloadCodeBaseForAdapterId()

Load or reload adapter code base

Param	ParamType	ParamValue	ParamDescription
customerID	int	<input type="text" value="1"/>	Customer id
adapterId	java.lang.String	<input type="text" value="ServiceDeskAdapter"/>	Adapter id

- 6 Click the **Invoke** button below the function. A success window opens.



JMX MBean Operation Result

loadOrReloadCodeBaseForAdapterId()

[Back to Agent View](#) [Back to MBean View](#) [Reinvoke MBean Operation](#)

load or reload was succesful !!!


Create a web service object to support the CI attribute

UCMDB uses the Service Manager `ucmdbIntegration` web service to send CI data. This web service publishes the objects that match the out-of-box CI types and CI attributes provided by the UCMDB integration. See [Managed fields](#) on page 44 for a list of the out-of-box web service objects and their mappings to Service Manager tables.

If you add a CI attribute to the integration on your UCMDB system, you must create a corresponding web service object on your Service Manager system to receive the incoming CI data. Each web service object must map to a valid Service Manager table and column.

The following steps illustrate how to create a web service object for the OSVendor attribute described in previous sections.

To create a web service object for the CI attribute:

- 1 Log in to Service Manager with an administrator account.
 - 2 Click **Tailoring > Web Services > WSDL Configuration**. The External Access Definition form opens.
 - 3 For Service Name, type **ucmdbIntegration**.
 - 4 Click **Search**. Service Manager displays a record list of the objects that make up the `ucmdbIntegration` web service.
 - 5 Select the existing web service object to which you want to add the CI attribute. For example, select **ucmdbComputer**.
 - 6 Click the **Fields** tab. Service Manager displays the fields published as web service objects.
 - 7 Select an empty row in the Fields list.
 - 8 For Field, select the Service Manager column name where you want to store the incoming CI attribute values. For example, **os.manufacturer**.
-  Service Manager displays the fields from all join tables associated with the table listed in the Name field. For example, `joincomputer` displays fields from the device and computer tables.

- 9 For Caption, type the name you want Service Manager to use when publishing the field as a web service object. For example, **OSVendor**.

▶ The Caption name must match the object name you listed in the XSL transformation file on the UCMDB.

- 10 Click **Save**. The new web service object will be available the next time you restart the Service Manager system.

External Access Definition

Service Name: Released

Name: Depreciated

Object Name:

◆ Allowed Actions ◆ Expressions ◆ Fields

Field	Caption	Type
ucmdb.id	UCMDBId	
default.gateway	DefaultGateway	
operating.system	OS	
os.version	OSVersion	
bios.id	BIOSId	
network.name	DNSName	
addIPAddr[addl...	AddIPAddress	
addIPAddr[addl...	AddSubnet	
addMacAddress	AddMacAddress	
physical.mem.total	PhysicalMemory	
subtype	Subtype	
building	Building	
room	Room	
floor	Floor	
machine.name	HostName	
os.manufacturer	OSVendor	

Add a managed field to support the CI attribute

In order for a CI attribute you add to the integration to trigger the automated Change Management validation and verification processes, you must add a managed field for the CI attribute. Service Manager managed fields are part

of the Discovery Event Manager Rules. See [Service Manager Discovery Event Manager rules](#) on page 50 for a list of fields that trigger Change Management validation and verification.

To add a managed field to the integration:

- 1 Log in to Service Manager with an administrator account.
- 2 Click **Tailoring > Web Services > Discovered Event Manager Rules**. Service Manager displays the record search form.
- 3 Click Search to display a list of all Discovery Event Manager rules.
- 4 Select the rule ID that matches the web services object where you mapped the incoming CI attribute. For example, **ucmdbComputer**. Service Manager displays the rules for this web service object.
- 5 Click the **Managed Fields** tab. Service Manager displays the list of fields that trigger Change Management validation and verification.
- 6 Select an empty row in the Managed Fields list.
- 7 For Field Name, select the caption name of the Service Manager column that you previously selected to store the incoming CI attribute values. For example, **Os Manufacturer**.
 - ▶ Service Manager displays the fields from all join tables associated with the table listed in the Table Name field. For example, **joincomputer** displays fields from the device and computer tables.
- 8 If the field you use to store the incoming CI attribute is an array of structure, use the Structure field to select the name of the array of structure where the column can be found. For example, Os Manufacturer is a primitive character field and therefore does not need to identify an array of structure name.
- 9 If the field you use to store the incoming CI attribute is an array of structure, use the Index field to select the index number that identifies the column in the array of structure. For example, Os Manufacturer is a primitive character field and therefore does not need to identify an array of structure index.
- 10 Click **Save**. The managed field will be available the next time you restart the Service Manager system.

Discovery Event Manager Rules

Id:
Table Name:
Condition:

Field Name	Structure	Index	Condition
Operating System			
Default Gateway			
Os Version			
Bios Id			
Network Name			
addIPAddress	addIPAddr	1	
addSubnet	addIPAddr	2	
Add Mac Address			
Physical Mem Total			
Machine Name			
Subtype			
Building			
Room			
Floor			
Os Manufacturer			

Add a CI Type to the integration

You can use the following steps to add a CI type to the integration.

Task 1: Does the CI type already exist in the UCMDB class model?

Yes. Go to [Task 3](#).

No. Go to [Task 2](#).

Task 2: Add the CI type to the UCMDB class model.

See [Add a CI type to the UCMDB class model](#) on page 77.

Task 3: Add CI attributes to the CI type as needed.

See [Add a CI attribute to the integration](#) on page 59.

Task 4: Create a TQL query to synchronize the CI type's attributes.

See [Create a TQL query to synchronize the CI type](#) on page 81.

Task 5: Verify all relationships in the TQL query have the tracking qualifier.

See [Verify all relationships in the TQL query have the tracking qualifier](#) on page 85.

Task 6: Enable calculation for the CI type's attributes.

See [Enable calculation for the CI type's attributes](#) on page 86.

Task 7: Map the CI type's TQL query to an XSL transformation file.

See [Map the CI type's TQL query to an XSL transformation file](#) on page 89.

Task 8: Map the CI type's attributes to web service objects

See [Map the CI type's attributes to web service objects](#) on page 92.

Task 9: Reload the code base for the ServiceDeskAdapter.

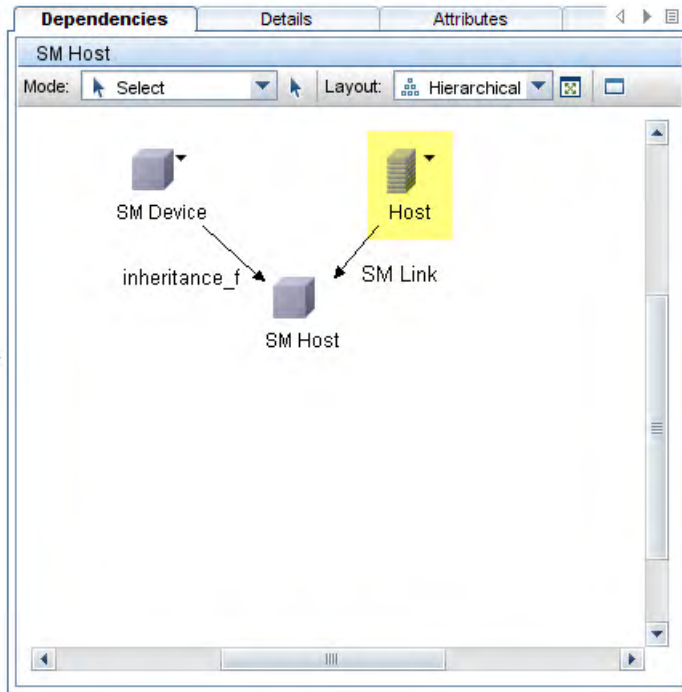
See [Reload the code base for the ServiceDeskAdapter](#) on page 97.

- Task 10: Add the CI type's TQL query to the integration Changes Adapter.
See [Add custom TQL queries to the integration Changes Adapter](#) on page 99.
- Task 11: Add the CI type's TQL query to the integration replication jobs.
See [Add custom TQL queries to integration replication jobs](#) on page 100.
- Task 12: Create web service objects to support the CI type.
See [Create web service objects to support the CI type](#) on page 101.
- Task 13: Add managed fields to support the CI type.
See [Add managed fields to support the CI type](#) on page 103.

Add a CI type to the UCMDB class model

The integration class model only uses a subset of the CI types available from your UCMDB system. Out-of-box, the integration consists of CI types that are typically managed from a Service Manager system such as hosts and network components. Before creating a new UCMDB CI type, you should determine if there are any existing CI types in your UCMDB system that provide the CI attributes you want. In most cases, you can create links to one or more existing CI types to create a new logical CI type for use by the integration.

For example, the SM Host CI is a logical CI that links to the Host CI. The SM Host CI inherits CI attributes from Host and all of its child CI types.





The following steps illustrate how to create a new CI type called SM RDBMS based on an existing CI type called database.

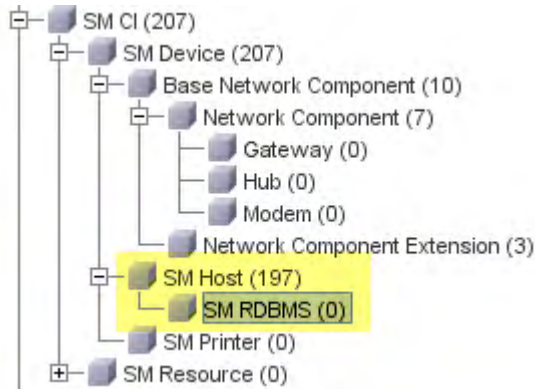


The integration does not require any special steps to add a CI type to the UCMDB class model. You can use the standard CI type creation procedures to add a CI type. For more information on CI type creation, see the *HP Universal CMDB CI Attribute Customization Guide*.

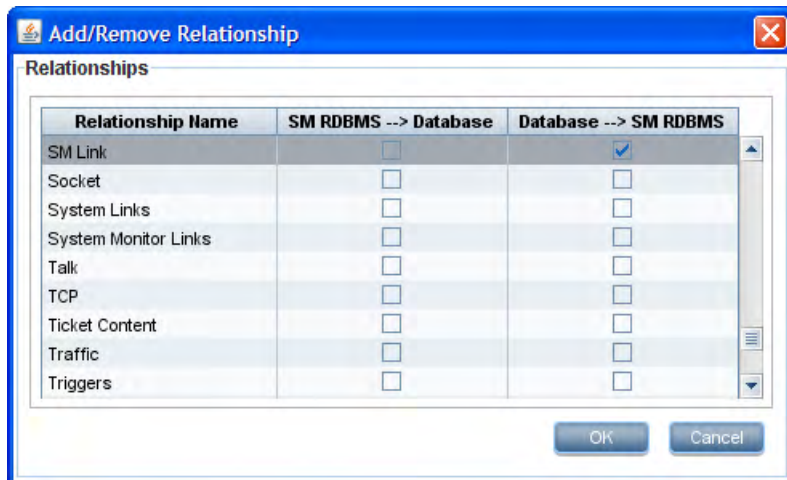
To add a CI type to the UCMDB class model:

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the **Modeling** tab.
- 3 Click **CI Type Manager**.
- 4 Select the base CI type you want to use for your new CI type from the CI Types navigation tree. For example, **Data > Object > SM CI > SM Device > SM Host**.

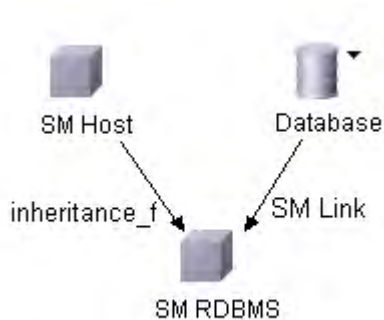
- 5 Click the new button. 
The Create Configuration Item Type window opens.
- 6 In Name, type the unique name you want to use for the new CI type. For example, **sm_rdbms**.
 The name cannot include any of the following characters: ` / \ [] : | < > + = ; , ? *.
- 7 In Display Name, type the name you want UCMDB to display in the interface. For example, **SM RDBMS**.
- 8 In Description, type a description of the new CI type. This is an optional field. For example, **Hosts running relational databases**.
- 9 In Base CI Type, verify that the proper base CI type is selected. Your new CI type will inherit the attributes of the base CI type you select here. For example, **SM Host**.
- 10 Click **Next**. The wizard displays the list of CI attributes from the base CI type.
- 11 Add, edit, or remove CI attributes as needed for the new CI type. For example, accept the default attributes inherited from SM Host.
- 12 Click **Next**. The wizard displays the list of qualifiers from the base CI type.
- 13 Add or remove qualifiers as needed for the new CI type. For example, accept the default qualifiers.
- 14 Click **Next**. The wizard displays the list of icons associated with the CI type.
- 15 Select the icons associated with this CI type. For example, accept the default abstract class icon.
- 16 Click **Next** to add any menu item properties or label definitions as needed. For example, accept the default settings from the base CI type.
- 17 Click **Finish** to create the CI type.




- 18 Select your new CI type from the tree. For example, **SM RDBMS**.
- 19 Browse to an existing CI type you want to link to, and control-click it to add it to your selection. For example **Database**.
 - ▶ Choose an existing CI type that has the attributes that you want to be part of your new logical CI type.
- 20 Right-click one of the selected CI types, and click **Add/Remove Relationship**. The Relationships window opens.
- 21 Create an **SM Link** relationship from the existing CI type to the new CI type. For example, from **Database** to **SM RDBMS**.



22 Click **OK** to create the relationship.




23 Click the save button to save the CI type. 

Create a TQL query to synchronize the CI type

The integration uses Topology Query Language (TQL) queries to gather CI attribute values and pass them to your Service Manager system. You must create a TQL query for any CI type you add to the integration. Any TQL query you create must conform to the [TQL query requirements](#) on page 43.

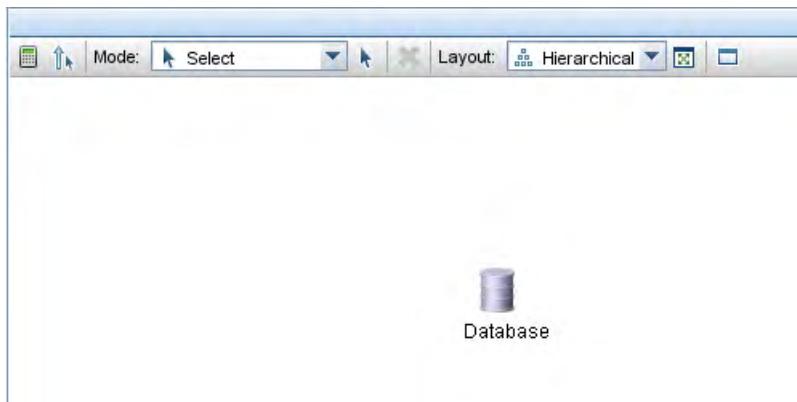
The following steps illustrate how to create a new TQL query called `rdbmsData` for the SM RDBMS CI type described in previous sections.

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the **Modeling** tab.
- 3 Click **Query Manager**.
- 4 From the Queries navigation tree, click **Integration > SM Sync**.
- 5 Click the new button. 

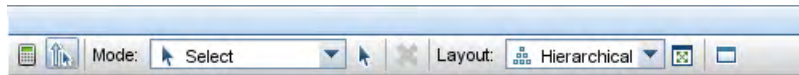
The Create New Query window opens.

- 6 In Name, type the unique name you want to use for the new query. The name cannot include any of the following characters: ` / \ [] : | < > + = ; , ? *. For example, **rdbmsData**.

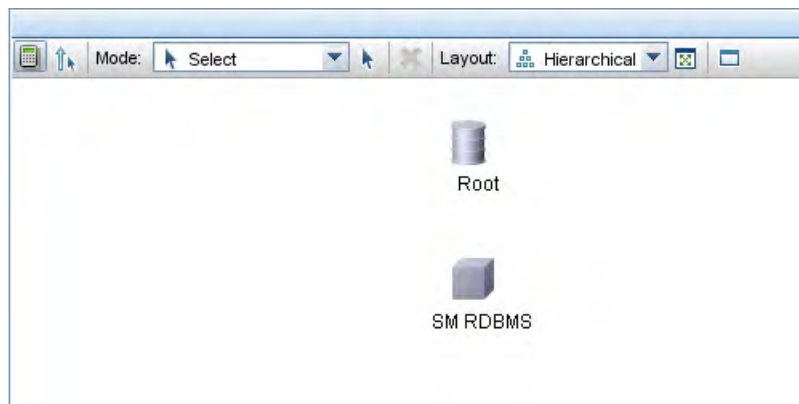
- 7 In Description, type a description of the new query. This is an optional field. For example, **Query for hosts running relational databases**.
- 8 For Query Type, select **Integration**. The integration will only synchronize CI types that have a query from this folder.
- 9 For Query Priority, select how frequently you want the system to rerun the query. For example, select **Medium** to accept the default priority.
- 10 Enable or disable persistence as needed. If you want the TQL permanently remain in system memory, select persistent. For example, accept the default state with persistence disabled.
- 11 Click **OK**. UCMDB adds your new query to the Queries list.
- 12 Find the CI type that will be the root node of your query from the CI Type Selector. This CI type is typically the one that provides the most attributes for the CI. For example, **Data > Object > IT Universe > System > Software Element > Database**.
 - ▶ In this case, the **Database** CI type is a better root node element than the **SM RDBMS** CI type because it provides more CI attributes.
- 13 Drag the root CI type from the CI Type Selector and drop it into the empty Editing pane. UCMDB displays the icon of the CI type.



- 14 Select the CI type, and click **Edit** from the Information Pane. The Node properties window opens.
- 15 Change the Element name to **Root**.
- 16 Click **OK** to save the node properties.

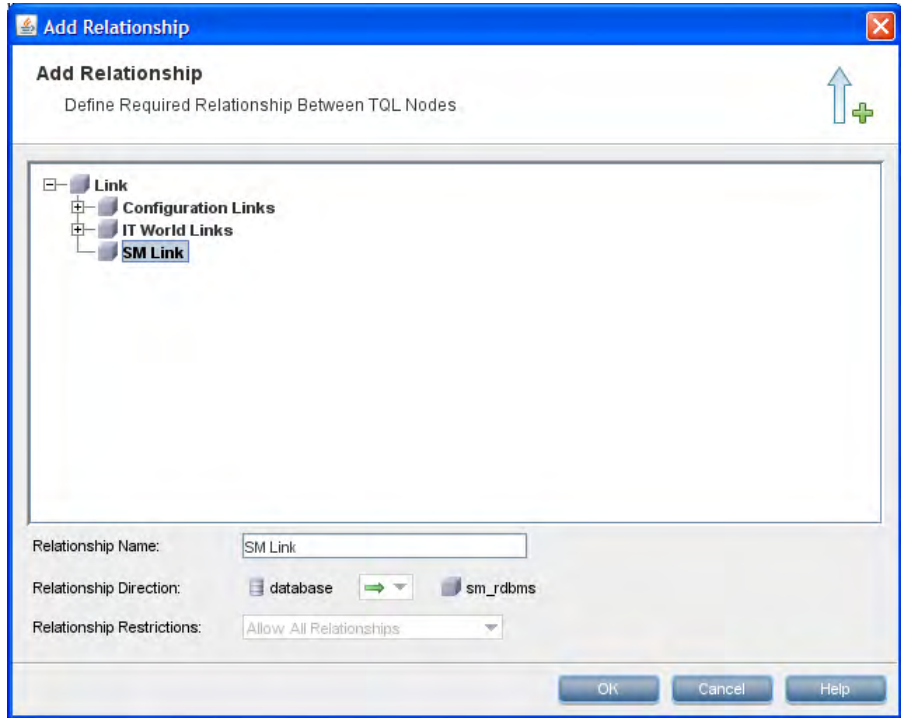


- 17 Find any additional CI types you want to add to the query from Type Selector. These CI types typically provide additional CI attributes. For example, **Data > Object > SM CI > SM Device > SM Host > SM RDBMS**.
- 18 Drag the additional CI type from the CI Type Selector and drop it into the empty Editing pane. UCMDB displays the icon of the additional CI type.




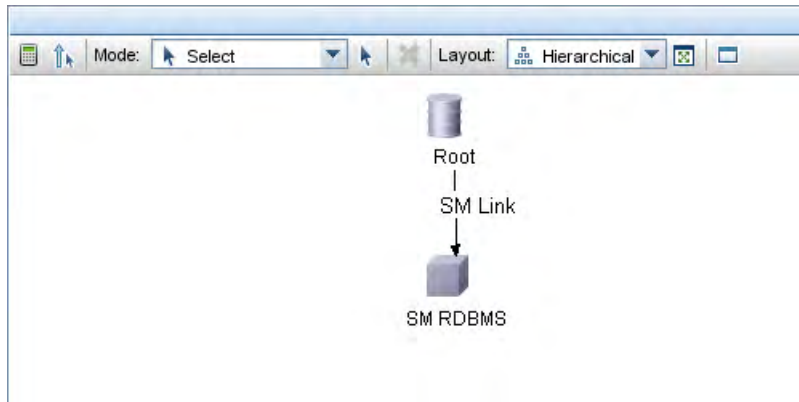
- 19 Create relationships between the Root CI type and the additional CI types as needed. For example, create an SM Link between **Root** and **SM RDBMS**.
 - a Select **Root** and control-click the additional CI type. For example, **SM RDBMS**.
 - b Right-click one of the selected items, and click **Add Relationship**. The Add Relationship window opens.
 - c Select SM Link.

- d Type a Relationship Name. For example **SM Link**.
- e Verify that the relationship direction is from the root CI type to the additional CI type. For example, from database to sm_rdbms.
- f Click **OK** to add the relationship.



20 Repeat [step 17](#) to [step 19](#) for each additional CI type you want to add to the TQL. For example, SM RDBMS does not need any additional CI types.

21 Click the save button to save the TQL query. 

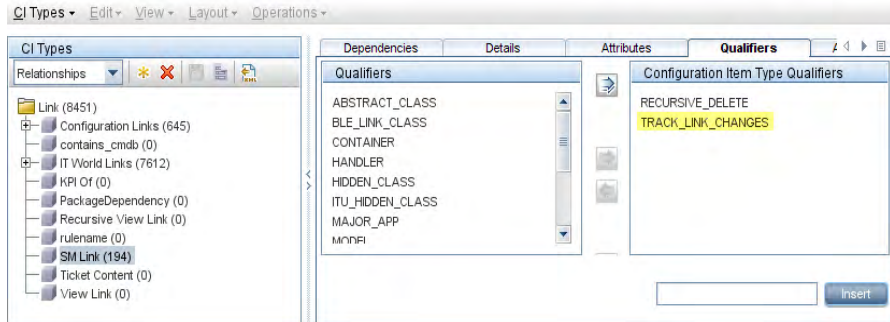



Verify all relationships in the TQL query have the tracking qualifier

Most TQL queries you create will consist of one or more relationships between CI types. In order for the integration to track the history of these CI relations, you must add the `TRACK_LINK_CHANGES` qualifier to each relationship in the TQL query. You should verify that each relationship in your TQL query has the `TRACK_LINK_CHANGES` qualifier so that the integration will receive updates for the relation.

The following steps illustrate verifying that the SM Link relationship has the `TRACK_LINK_CHANGES` qualifier.

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 From the Modeling tab menu, select **CI Type Manager**.
- 3 From CI Types, select **Relationships**.
- 4 Select the Relationship you want to edit from the navigation tree. For example, **Link > SM Link**.
- 5 Click the **Qualifiers** tab.
- 6 Verify that the Configuration Item Type Qualifiers column contains the `TRACK_LINK_CHANGES` qualifier.
- 7 If the relationship does not already contain it, select **TRACK_LINK_CHANGES** from the Qualifiers list, and click the right arrow to move it over to the Configuration Item Type Qualifiers section.



- 8 Click the save button to save the CI relationship. 

Enable calculation for the CI type's attributes

To add a CI attribute to the integration, you must enable the calculation layout setting from the TQL query that synchronizes the CI type. Because you must enable calculation for each attribute you want to add to the integration, you should be familiar with the integration CI types and the CI attributes they contain.




Keep a list of the attributes you enable, because you will need to create a matching XSL transformation for each one.

The following steps illustrate how to enable calculation for attributes of the SM RDBMS CI type described in previous sections.

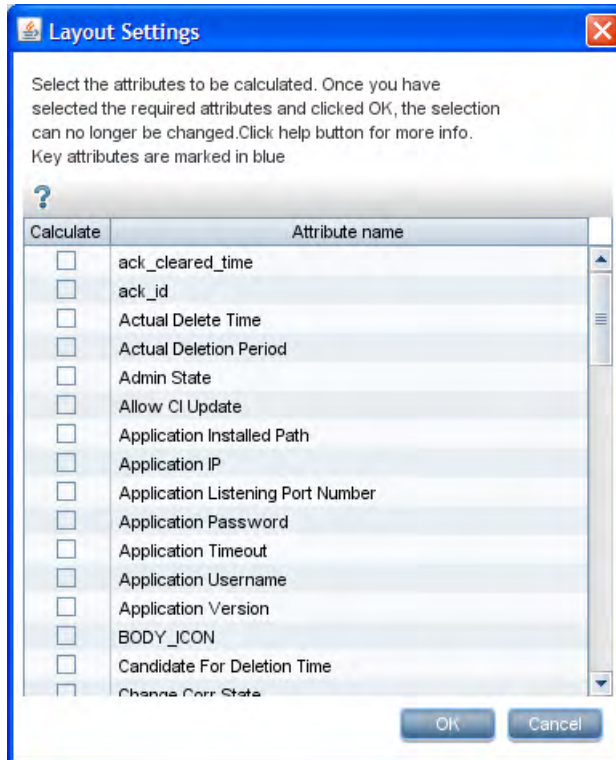
To enable calculation for a CI type's attributes

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the Modeling tab.
- 3 Click **Query Manager**.
- 4 From the Queries navigation tree, click **Integration > SM Sync**.
- 5 Select the query that manages the CI type whose attributes you want to add to the integration. For example, **rdbmsData**. UCMDB displays the TQL for the integration query.

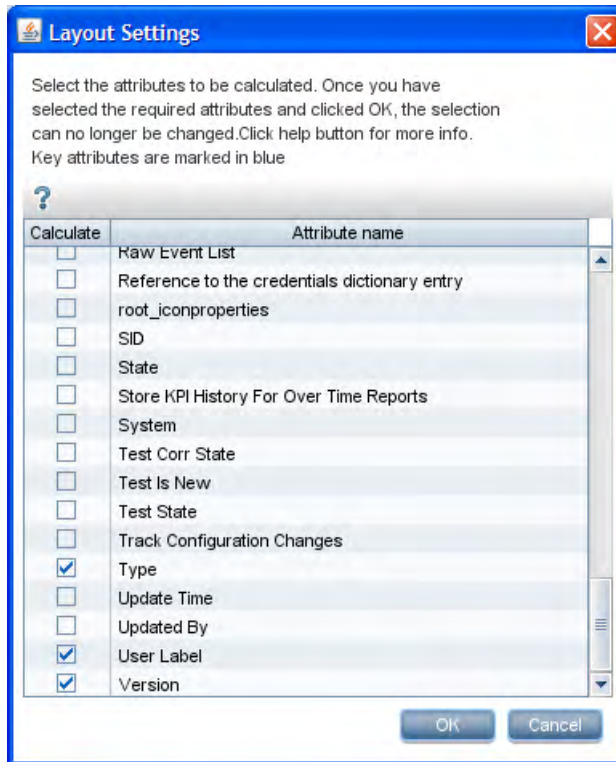
- 6 Select the **Root** node from the TQL layout, and then click the **Edit** button from the Information Pane. The Node properties window opens.

 Your integration query must contain a node called Root. See [TQL query requirements](#) on page 43 for more information.

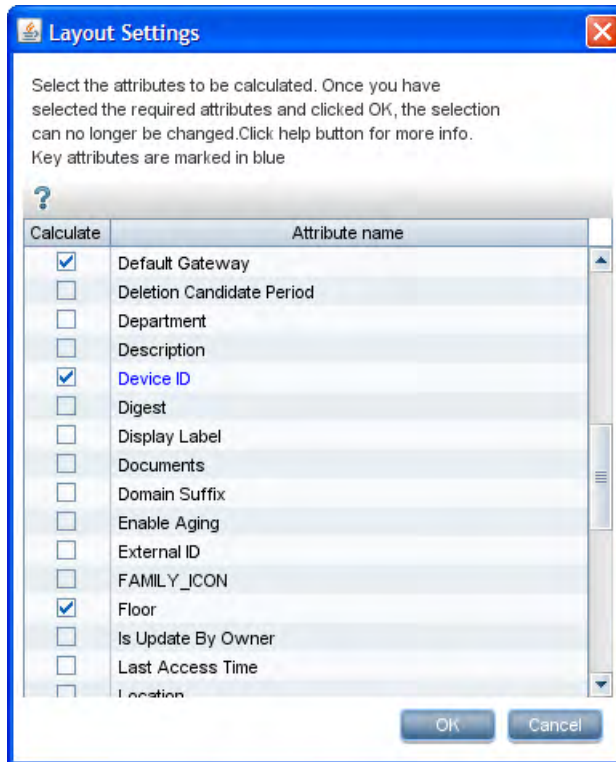
- 7 Click **Advanced layout settings**. The Layout Settings window opens.




- 8 Enable the Calculate option for each CI attribute you want to include in the integration. For example, enable calculation for the Application IP, Name, Type, User Label, and Version attributes.



- 9 Click **OK** to close the advanced layout settings window.
- 10 Click **OK** to save the node properties.
- 11 Select any additional nodes that contain CI attributes you want to add to the integration. For example, **SM RDBMS**.
- 12 Click the **Edit** button from the Information Pane. The Node properties window opens.
- 13 Click **Advanced layout settings**. The Layout Settings window opens.
- 14 Enable the Calculate option for each additional CI attribute you want to include in the integration. For example, enable calculation for the Building, Default Gateway, Device ID, Floor and Room attributes.



- 15 Click **OK** to close the advanced layout settings window.
- 16 Click **OK** to save the node properties.
- 17 Repeat [step 11](#) to [step 16](#) for each additional node that contains CI attributes that you want to add to the integration.
- 18 Click the save button to save the TQL query. 

Map the CI type's TQL query to an XSL transformation file

The integration uses a configuration file called `SyncConfFile.xml` to map each UCMDB TQL query to an XSL transformation file. In order for custom TQL queries to be part of the integration, you must add a mapping entry for each TQL query in the configuration file.

The following steps illustrate mapping the TQL query `rdbmsData` described in previous sections to the Service Manager `ucmdbApplication` web service.

To map a TQL query to an XSL transformation file

- 1 Log in to the UCMDB file system with an administrator account.
- 2 Browse to the `SyncConfFile.xml` path in the file system. For example:
**C:\hp\UCMDB\UCMDBServer\j2f\fcmdb\CodeBase
\ServiceDeskAdapter**
- 3 Use a text editor to open the `SyncConfFile.xml` configuration file.
- 4 Copy an existing TQL mapping element. A TQL mapping element uses the following format:

```
<tql name="TQL_query" xslFile="XSL_File">  
  <!-- Description of mapping -->  
  <request type="Create" name="Create_web_service"/>  
  <request type="Update" name="Update_web_service"/>  
  <request type="Delete" name="Delete_web_service"/>  
</tql>
```

TQL_query is the name of the UCMDB TQL query you created.

XSL_File is the name XSL transformation file the integration will use to map UCMDB attributes to Service Manager web service objects.

Create_web_service is the name of the Service Manager web service you want to the integration to use to create CIs from this TQL query.

Update_web_service is the name of the Service Manager web service you want to the integration to use to update CIs in this TQL query.

Delete_web_service is the name of the Service Manager web service you want to the integration to use to delete CIs from this TQL query.

Figure 5 Excerpt of smSynchConfFile.xml

```
- <config>
- <global-config>
  <request type="Create" cmdb-id-only="false" />
  <request type="Update" cmdb-id-only="false" />
  <request type="Delete" cmdb-id-only="true" />
</global-config>
- <mapping>
- <tql name="hostData" xslFile="host_data.xslt">
  <!-- this is host->ip,interface,sm_server tql -->
  <request type="Create" name="CreateucmdbComputerRequest" />
  <request type="Update" name="UpdateucmdbComputerRequest" />
  <request type="Delete" name="DeleteucmdbComputerRequest" />
</tql>
- <tql name="applicationData" xslFile="application_data.xslt">
  <!-- this is logical_application tql -->
  <request type="Create" name="CreateucmdbApplicationRequest" />
  <request type="Update" name="UpdateucmdbApplicationRequest" />
  <request type="Delete" name="DeleteucmdbApplicationRequest" />
</tql>
- <tql name="businessServiceData" xslFile="business_service_data.xslt">
  <!-- this is business_service_for_catalog tql -->
  <request type="Create" name="CreateucmdbBusinessServiceRequest" />
  <request type="Update" name="UpdateucmdbBusinessServiceRequest" />
  <request type="Delete" name="DeleteucmdbBusinessServiceRequest" />
</tql>
```

- 5 Add or update TQL mapping elements for each TQL query you want to add to the integration. For example, create the following TQL mapping between the rdbmsData TQL query and the rdbms_data.xslt file.

```
<tql name="rdbmsData" xslFile="rdbms_data.xslt">
  <!-- this is database tql -->
  <request type="Create"
  name="CreateucmdbApplicationRequest" />
  <request type="Update"
  name="UpdateucmdbApplicationRequest" />
  <request type="Delete"
  name="DeleteucmdbApplicationRequest" />
</tql>
```

```

- <config>
- <global-config>
  <request type="Create" cmdb-id-only="false" />
  <request type="Update" cmdb-id-only="false" />
  <request type="Delete" cmdb-id-only="true" />
</global-config>
- <mapping>
- <tql name="hostData" xslFile="host_data.xslt">
  <!-- this is host->ip,interface,sm_server tql -->
  <request type="Create" name="CreateucmdbComputerRequest" />
  <request type="Update" name="UpdateucmdbComputerRequest" />
  <request type="Delete" name="DeleteucmdbComputerRequest" />
</tql>
- <tql name="applicationData" xslFile="application_data.xslt">
  <!-- this is logical_application tql -->
  <request type="Create" name="CreateucmdbApplicationRequest" />
  <request type="Update" name="UpdateucmdbApplicationRequest" />
  <request type="Delete" name="DeleteucmdbApplicationRequest" />
</tql>
- <tql name="rdbmsData" xslFile="rdbms_data.xslt">
  <!-- this is database tql -->
  <request type="Create" name="CreateucmdbApplicationRequest" />
  <request type="Update" name="UpdateucmdbApplicationRequest" />
  <request type="Delete" name="DeleteucmdbApplicationRequest" />
</tql>

```

6 Save the configuration file.

Map the CI type's attributes to web service objects

The integration uses the Service Desk Adapter to transform UCMDB CI attributes to web services objects recognized by Service Manager. The Service Desk Adapter uses XSL transformation files to convert UCMDB TQL queries into a properly formatted Service Manager web services messages. Out-of-box, each integration query has a corresponding XSL transformation file. In addition, each attribute you enable for synchronization from Advanced layout settings requires its own entry in the XSL transformation file.

If you want to add a CI type to the integration, you must create a matching XSL transformation file that defines how the Service Desk Adapter transform each CI type into a Service Manager web service object. See [Integration TQL queries](#) on page 39 for information about which CI types each query manages. In order to create a proper XSL mapping, you must be familiar with the

service and object names Service Manager publishes as Web services. See the Service Manager help for information on publishing tables and columns as Web service objects.

The following steps illustrate creating an XSL transformation file for the rdbmsData TQL query described in previous sections.

To map a CI type's attributes to web service objects:

- 1 Log in to the UCMDB file system with an administrator account.
- 2 Browse to the Service Desk Adapter path in the file system. For example:
**C:\hp\UCMDB\UCMDBServer\j2f\fcmdb\CodeBase
\ServiceDeskAdapter**
- 3 Copy an existing XSL transformation file to use it as a template to create a new transformation file. For example, copy `application_data.xslt`.
- 4 Rename the new transformation file to a unique name. For example, `rdbms_data.xslt`.
- 5 Use a text editor to open the new XSL transformation file.
- 6 Find the CI type definition element. The CI type definition element uses the following format:

```
<xsl:template match="/CI_type_name">
```

`@CI_type_name` is the name of CI type in the UCMDB system.

Figure 6 CI type definition in the `application_data.xslt`

```
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="/logical_application">
- <model>
  <keys />
  <instance>
    <!-- import:cmdb_root_attributes.xslt -->
    <xsl:for-each select="@friendlyType">
      <Subtype>
        <xsl:value-of select="." />
      </Subtype>
    </xsl:for-each>
    <xsl:for-each select="@data_name">
      <ApplicationName>
        <xsl:value-of select="." />
      </ApplicationName>
    </xsl:for-each>
  </instance>
</model>
</xsl:template>
</xsl:stylesheet>
```

- 7 Update the CI type name to match the CI type you want to add to the integration. For example, create the following CI type definition element to add the database CI type to the integration.

```
<xsl:template match="/database">
```

- 8 Add or update table naming elements as needed. By default, UCMDB sends CI attribute data to the Service Manager device table. If you want to send CI attributes to one of the join tables of device, you must add an element to specify the table name using the format `<file.table_name>`. For example, you do not need to specify an additional jointable to define a database CI type since Service Manager does not use a separate jointable to manage database CI types.
- 9 Find the elements that transform UCMDB CI attributes into Service Manager web service objects. The CI attribute transformation elements use the following format:

```
<xsl:for-each select="@CI_attribute_name">
  <ObjectName><xsl:value-of select="." /></ObjectName>
</xsl:for-each>
```

`@CI_attribute_name` is the name of attribute in the UCMDB system.

`ObjectName` is the name of a web service object published by the Service Manager system.

Figure 7 CI attributes in application_data.xslt

```

- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="/logical_application">
- <model>
  <keys />
  <instance>
    <!-- import:ucmdb_root_attributes.xslt -->
    <xsl:for-each select="@friendlyType">
      <Subtype>
        <xsl:value-of select="." />
      </Subtype>
    </xsl:for-each>
    <xsl:for-each select="@data_name">
      <ApplicationName>
        <xsl:value-of select="." />
      </ApplicationName>
    </xsl:for-each>
  </instance>
</model>
</xsl:template>
</xsl:stylesheet>

```

Figure 8 Matching CI attributes in the ucmdbApplication web service

External Access Definition

Service Name: Released

Name: Deprecated

Object Name:

Allowed Actions
 Expressions
 Fields

Field	Caption	Type
ucmdb.id	UCMDBId	
ci.name	ApplicationName	
subtype	Subtype	

- 10 Add or update CI attribute transformation elements for each CI attribute you want to add to the integration. For example, create the following XSL transformation elements for the database CI type.

Table 15 Sample XSL transformation elements for database CIs

UCMDB attribute	Sample transformation elements
data_name	<pre data-bbox="615 366 1158 505"><xsl:for-each select="@data_name"> <ApplicationName><xsl:value-of select="."/></ApplicationName> </xsl:for-each></pre>
application_ip	<pre data-bbox="615 539 1236 635"><xsl:for-each select="@application_ip"> <DBIP><xsl:value-of select="."/></DBIP> </xsl:for-each></pre>
database_dbtype	<pre data-bbox="615 670 1250 800"><xsl:for-each select="@database_dbtype"> <DBType><xsl:value-of select="."/></ DBType> </xsl:for-each></pre>
database_dbversion	<pre data-bbox="615 835 1236 999"><xsl:for-each select="@database_dbversion"> <DBVersion><xsl:value-of select="."/></ DBVersion> </xsl:for-each></pre>
user_label	<pre data-bbox="615 1034 1236 1164"><xsl:for-each select="@user_label"> <UserLabel><xsl:value-of select="."/></ UserLabel> </xsl:for-each></pre>

Figure 9 New attribute mappings in rdbms_data.xslt

```
- <xsl:for-each select="@data_name">
- <ApplicationName>
  <xsl:value-of select="." />
</ApplicationName>
</xsl:for-each>
- <xsl:for-each select="@application_ip">
- <DBIP>
  <xsl:value-of select="." />
</DBIP>
</xsl:for-each>
- <xsl:for-each select="@database_dbtype">
- <DBType>
  <xsl:value-of select="." />
</DBType>
</xsl:for-each>
- <xsl:for-each select="@database_dbversion">
- <DBVersion>
  <xsl:value-of select="." />
</DBVersion>
</xsl:for-each>
- <xsl:for-each select="@user_label">
- <UserLabel>
  <xsl:value-of select="." />
</UserLabel>
</xsl:for-each>
```

- 11 Save the XSL transformation file.

Reload the code base for the ServiceDeskAdapter

In order for the integration to use your custom TQL queries and XSL transformation files, you must reload the code base of the ServiceDeskAdapter.

To reload the code base for the ServiceDeskAdapter:

- 1 Open a web browser and browse to the following URL:

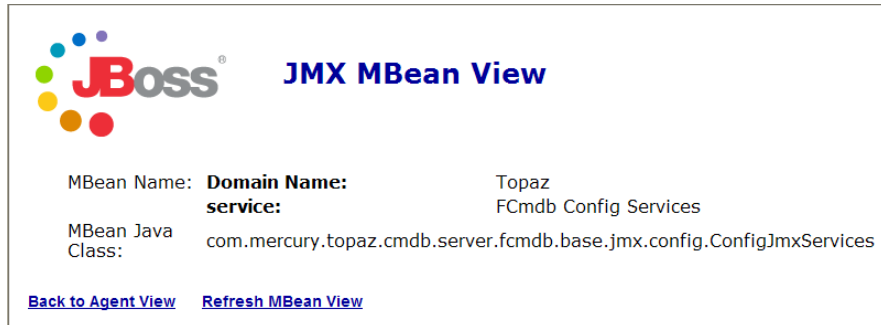
**http://<UCMDB server>:8080/jmx-console/
HtmlAdaptor?action=inspectMBean&name=Topaz%3AService%3
DFCmdb+Config+Services**

UCMDB server is the fully qualified domain name for your UCMDB system.

- 2 When prompted, provide your UCMDB administrator user name and password.

The JMX MBean View web page opens.

Figure 10 JMX MBean View



- 3 Scroll to the section for the loadOrReloadCodebaseForAdapterId() MBean.
- 4 For customerID, type a ParamValue of **1**.
- 5 For adapterId, type a ParamValue of **ServiceDeskAdapter**.

java.lang.String loadOrReloadCodeBaseForAdapterId()

Load or reload adapter code base

Param	ParamType	ParamValue	ParamDescription
customerID	int	1	Customer id
adapterId	java.lang.String	ServiceDeskAdapter	Adapter id

- 6 Click the **Invoke** button below the function. A success window opens.



JMX MBean Operation Result

loadOrReloadCodeBaseForAdapterId()

[Back to Agent View](#) [Back to MBean View](#) [Reinvoke MBean Operation](#)

load or reload was succesful !!!

Add custom TQL queries to the integration Changes Adapter

In order for the integration to update the values of your custom CI types and attributes, you must add your custom TQL queries to the Changes adapter. The following steps illustrate adding the `rdbmsData` TQL query described in the previous sections to the Changes Adapter.

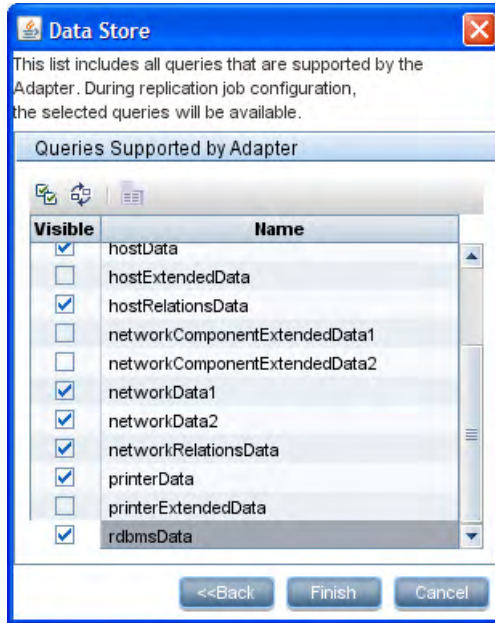
To add custom TQL queries to the Changes Adapter

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the Settings tab.
- 3 Click the Data Stores tab.
- 4 Select the Changes Adapter you created to support the integration. For example, **cmdbChanges**.

- 5 Click the Edit button. 

UCMDB displays the Data Store properties window.

- 6 Click Next to go to the Queries Supported by the Adapter page.
- 7 Enable the Visible option for your custom TQL query. For example, **rdbmsData**.




- 8 Click **Finish** to update the Changes Adapter.

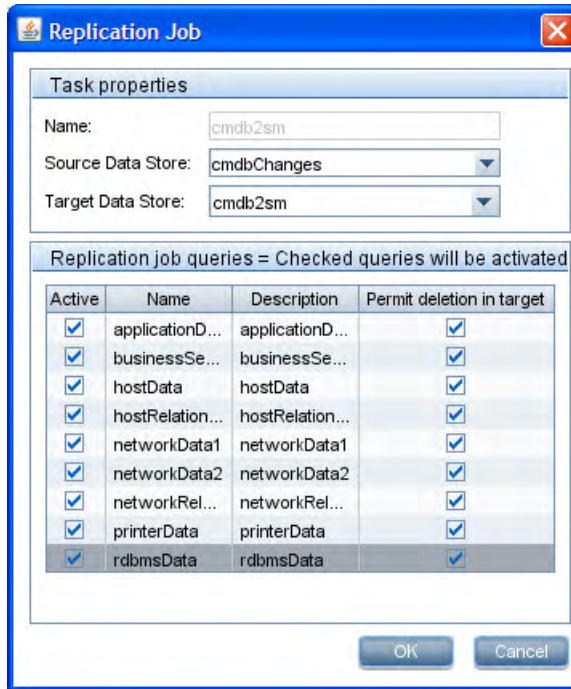
Add custom TQL queries to integration replication jobs

In order for the integration to send your custom CI types and attributes to your Service Manager system, you must add your custom TQL queries to the replication job between your Changes data store and your Service Manager data store. The following steps illustrate how to add the custom rdbmsData TQL query described in the previous sections.

To add custom TQL queries to replication jobs

- 1 Log in to your HP Universal CMDB system as an administrator.
- 2 Click the Settings tab.
- 3 Click the Replication Jobs tab.
- 4 Select the replication job you created to synchronize CIs between your Changes data store and your Service Manager data store. For example, **cmdb2sm**.

- Click the Edit button. 
- UCMDB displays the Replication Job Task properties window.
- Enable the Active and Permit deletion in target options for your custom TQL query. For example, **rdbmsData**.



- Click **OK** to update the replication job.


Create web service objects to support the CI type

In order to add a CI type to the integration, you must create a Service Manager web service object for each CI attribute for which you created an XSL transformation on the UCMDB system. Service Manager uses the web service object to determine which Service Manager table and column to store the incoming CI attribute values.

The following steps illustrate how to create the web service objects necessary to support the SM RDBNS CI type described in previous sections.

To create web service objects to support your new CI type

- 1 Log in to Service Manager with an administrator account.
- 2 Click **Tailoring > Web Services > WSDL Configuration**. Service Manager displays the record search form.
- 3 In ServiceName, type **ucmdbIntegration** and click **Search**. Service Manager displays all the web service objects that are part of the integration.
- 4 Select the web service object you want to use to manage the incoming CI data. For example, **ucmdbApplication**.
 - ▶ Select the web service object that uses the Service Manager table where you want to store the incoming CI attributes. For example, **ucmdbApplication** publishes columns from the Service Manager device table and is a good out-of-box fit for RDBMS CIs.
- 5 Click the Fields tab.
- 6 Select an empty row in the Fields list.
- 7 For Field, select the Service Manager you want to use to store the incoming CI attribute data.
- 8 For Caption, type in the web service object name you used in the XSL transformation file.

 The web service object names must match the names you used in the XSL transformation file, or Service Manager will not receive any CI updates from your UCMDb system.

For example, type the following Field and Caption values to support the SM RDBMS CI type.

Table 16 Web service object definitions for the SM RDBMS CI type

Field	Caption
admin.urlport	DBIP
addl[addl.type]	DBType
product.version	DBVersion
addl[addl.name]	UserLabel

- 9 Click **Save**.

External Access Definition

Service Name: Released
Name: Depreciated
Object Name:

Allowed Actions Expressions **Fields**

Field	Caption	Type
ucmdb.id	UCMDBId	
ci.name	ApplicationName	
subtype	Subtype	
admin.urlport	DBIP	
add[addl.type]	DBType	
product.version	DBVersion	
add[addl.name]	UserLabel	

- 10 Stop and restart the Service Manager server. The new web service objects will now be available to the integration.

Add managed fields to support the CI type

In order for your custom CI type to trigger the automated Change Management validation and verification processes, you must add a managed field for each CI attribute within your CI type. Service Manager managed fields are part of the Discovery Event Manager Rules. See [Service Manager Discovery Event Manager rules](#) on page 50 for a list of fields that trigger Change Management validation and verification.

The following steps illustrate how to add the managed fields for the SM RDBMS CI type described in the previous sections.

To add managed field to support your CI type:

- 1 Log in to Service Manager with an administrator account.
- 2 Click **Tailoring > Web Services > Discovered Event Manager Rules**. Service Manager displays the record search form.

- 3 Click **Search** to display a list of all Discovery Event Manager rules.
- 4 Select the rule ID that matches the web services object where you mapped the incoming CI attribute. For example, **ucmdbComputer**. Service Manager displays the rules for this web service object.
- 5 Click the **Managed Fields** tab. Service Manager displays the list of fields that trigger Change Management validation and verification.
- 6 Select an empty row in the Managed Fields list.
- 7 For **Field Name**, select the caption name of the Service Manager column that you previously selected to store the incoming CI attribute values. For example, **Os Manufacturer**.
 - ▶ Service Manager displays the fields from all join tables associated with the table listed in the **Table Name** field. For example, **joincomputer** displays fields from the **device** and **computer** tables.
- 8 If the field you use to store the incoming CI attribute is an array of structure, use the **Structure** field to select the name of the array of structure where the column can be found. For example, **Os Manufacturer** is a primitive character field and therefore does not need to identify an array of structure name.
- 9 If the field you use to store the incoming CI attribute is an array of structure, use the **Index** field to select the index number that identifies the the column in the array of structure. For example, **Os Manufacturer** is a primitive character field and therefore does not need to identify an array of structure index.
- 10 Click **Save**.

Discovery Event Manager Rules

Id:

Table Name:

Condition:

Field Name	Structure	Index	Condition
Operating System			
Default Gateway			
Os Version			
Bios Id			
Network Name			
addIPAddress	addIPAddr	1	
addSubnet	addIPAddr	2	
Add Mac Address			
Physical Mem Total			
Machine Name			
Subtype			
Building			
Room			
Floor			
Os Manufacturer			

- 11 Stop and restart the Service Manager server. The new managed fields will now be available to the integration.