

HP Server Automation

Version: SA 9.14

SA-uCMDB Integration Guide

Software Release Date: December 17, 2012

Document Release Date: January 11, 2013



Contents

Document Information.....	3
The SA-uCMDB Integration.....	6
What's New	6
Installing and Configuring the SA-uCMDB Connector	7
Customizing SA Data Sent to the uCMDB Server.....	8
The Mapping File	8
Customizing the Mapping File	8
Editing the Mapping File.....	9
Support for SA Custom Attributes.....	12
Extended Out-Of-The-Box Mappings	13
Managing the SA-uCMDB Connector	16
Stopping and Disabling the SA-uCMDB Connector.....	16
The stop Command	16
The disable Command	16
Enabling and Starting the SA-uCMDB Connector	17
The enable Command.....	17
Displaying the Status of the SA-uCMDB Connector.....	18
SA-uCMDB Data Relationship and Transfer.....	19
CI Relationships Maintained.....	19
Example uCMDB Showing an SA Managed Server	19
SA Data Transferred to uCMDB	20
Frequency of Data Transfer to uCMDB.....	21
Accessing the uCMDB Browser from the SA Client	22
The uCMDB Browser Window	22
Configuring the uCMDB Browser.....	22
Support for uCMDB Server Versions 9.05 and 10.01	24
Global uCMDB IDs	24
SSL Connectivity to the uCMDB Server and the uCMDB Browser	24
Troubleshooting Tips.....	25
Running the SA-uCMDB Connector on a Second Core	25
On-demand Synchronization.....	26
Viewing Log Files	26
SA-uCMDB Connector Daemon	26
Appendix A – Mapping File.....	27

Document Information

In this section:

- [Legal Notices](#)
- [Documentation Updates](#)
- [Document Changes](#)
- [Support](#)
- [Support Matrices](#)

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 2000-2012 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

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://support.openview.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 can also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details. For a

Document Changes

The following table indicates changes made to this document since the last released edition.

Date	Changes
21 Dec 2012	Original version of the guide for SA 9.14, includes instructions for using the new functionality introduced in 9.14.
11 Jan 2013	<p>Moved document information to the front of the document.</p> <p>Page 9: Editing the Mapping File: Added to the permissions note that you need to log in to the SA Core as <i>root</i>.</p> <p>Page 10: XML Attribute Values:</p> <p>Corrected model-name value for <i>Model-Definition</i> to indicate that it is <i>NOT editable</i>.</p> <p>Fixed syntax error to remove space in the sample target-attr value for <i>Attribute-Custom</i>: changed <code>'serial number'</code> to <code>'serial_number'</code>.</p> <p>Page 12: Filter Support for Queries: Fixed incorrect value for enable command: changed <code>enable='false'</code> to <code>enable='true'</code>.</p> <p>Page 13-14: Customized Data Conversion Function: explained the customized text in the mapping file more.</p> <p>Page 17-18: Enabling and Starting the SA-uCMDB Connector: explained the enable command more. Modified the explanation of the new syntax. Corrected syntax in enable command examples.</p> <p>Page 22: Configuring the uCMDB Browser: Corrected parameter names in the examples for <code>--browser_protocol</code>, <code>--browser_host</code>, and <code>--browser_port</code>.</p>

Support

Visit the HP Software Support Online web site at: <http://support.openview.hp.com/>

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://support.openview.hp.com/access_level.jsp

Support Matrices

For complete support and compatibility information, see the support matrix for the relevant product release. All support matrices and product manuals are available here on the HP Software Support Online web site:

http://support.openview.hp.com/sc/support_matrices.jsp

You can also download the HP Server Automation Support and Compatibility Matrix for this release from the HP Software Support Online Product Manuals web site:

<http://support.openview.hp.com/selfsolve/manuals>

The SA-uCMDB Integration

This describes how to integrate HP Server Automation (SA) with the HP Universal Configuration Management Database (uCMDB) using the **SA-uCMDB Connector**. The SA-uCMDB Connector provides a single source for configuration data for asset compliance reporting.

HP SA stores a large amount of information about your servers and software in the SA database. The **SA-uCMDB Connector** copies some of this data to the HP uCMDB. Whenever the data in SA changes, the SA-uCMDB Connector automatically sends the updated data to the uCMDB Server.

The HP Universal CMDB is a configuration management database (CMDB) for enterprise IT organizations to document, store, and manage business service definitions and associated infrastructure relationships. The uCMDB provides a shared single version of truth to support business service management, IT service management, change management, and asset management initiatives. These initiatives help align IT efforts with business requirements and run IT operations more efficiently and effectively.

HP Server Automation provides life cycle management for enterprise servers and applications from discovery to provisioning, patching to configuration management and script execution to compliance assurance. HP Server Automation automates operations and processes across disparate IT teams and systems.

What's New

The SA-uCMDB Integration for SA 9.14 includes fixes to the SA-uCMDB Connector, as well as some key functionality enhancements.

Enhancements

- [Extended Out-Of-The-Box mappings](#)
- [Extensible ETL mapping and data normalizing capabilities](#)
- [Global uCMDB IDs](#)
- [On-demand Sync](#)
- [SSL Connectivity to the uCMDB Server and the uCMDB Browser](#)
- [Support for SA Custom Attributes](#)
- [Support for uCMDB server versions 9.05 and 10.01](#)

uCMDB Browser

With the SA-uCMDB Connector enabled in 9.14, the SA Client now provides the ability to launch the uCMDB Browser-Impact widget against an SA managed server.

For more information, see [Support for uCMDB Browser integration in the SA Client](#).

Installing and Configuring the SA-uCMDB Connector

The SA-uCMDB Connector is installed when you install SA. No separate installation is required.

If you are upgrading to Server Automation 9.14, the uCMDB Server must already be upgraded to release 9.05, 10.01 or later.

To download the latest Cumulative Update Package:

- 1 The SA 9.14 SA-uCMDB Connector requires that you run with uCMDB 9.05 or uCMDB 10.01 or later. uCMDB 9.05 includes Content Pack 10 and uCMDB 10.01 includes Content Pack 12; both are minimal requirements for using the SA-uCMDB Connector.

If you are running with uCMDB 9.05 you need to make sure it is running with the Cumulative Update Package 6 (CUP 6) or later.

The latest CUP HP Software Patch is available on the SSO Portal at the following locations:

— Windows:

http://support.openview.hp.com/selfsolve/document/FID/DOCUMENTUM_uCMDB_00094

— Linux:

http://support.openview.hp.com/selfsolve/document/FID/DOCUMENTUM_uCMDB_00095

For version support information, see [Support for uCMDB Server Versions 9.05 and 10.01](#).

- 2 Run the **enable** command to configure the SA-uCMDB Connector with the new uCMDB server.

NOTE: The syntax of the enable command varies depending on your environment. See [The enable Command](#) in this document for an explanation of the enable command syntax and options.

- 3 Enter the following command to start the SA-uCMDB Connector:

```
/etc/init.d/opsware-sas start telldaemon
```

- 4 Optionally check the status of the SA-uCMDB Connector with the following command:

```
/etc/init.d/opsware-sas status telldaemon
```

Customizing SA Data Sent to the uCMDB Server

The Mapping File

The SA-uCMDB Connector XML mapping file describes the data being transferred by the SA-uCMDB connector and enables you to customize the data mappings.

The initial mapping.xml is generated when the connector first runs. After it is generated, you can find the new mapping file at:

```
/etc/opt/opsware/tell/metadata/mapping.xml
```

The mapping file allows you to control:

- the data type and attributes that populate uCMDB and
- the mappings between the optional SA custom attributes and the uCMDB Data Model Configuration Item (CI) attributes.

NOTE: See [Appendix A – Mapping File](#) for the complete original mapping file contents.

Customizing the Mapping File

In order to customize how data is mapped, you need to edit the mapping.xml file. However, the mapping.xml file is not used by default; you need to restart the Connector to engage the customized mapping file.

To customize the uCMDB Connector mappings:

- 1 If the uCMDB Connector is running, you must stop and disable the Connector before editing the mapping file.

NOTE: See [Stopping and Disabling the SA-uCMDB Connector](#) for instructions.

IMPORTANT: Make sure the connector is stopped and disabled. If the connector is not stopped and disabled when you edit the mapping file, you may encounter problems when you try to restart the Connector.

- 2 Edit the `/etc/opt/opsware/tell/metadata/mapping.xml` as needed.

NOTE: See [Editing the Mapping File](#) for details on how to edit the mapping file for different purposes.

- 3 Run the **enable** command to change the configurations of the SA-uCMDB Connector.

NOTE: The syntax of the enable command varies depending on your environment. See [The enable Command](#) in this document for an explanation of the enable command syntax and options.

- 4 Run the **start** command to restart the uCMDB Connector:

```
/etc/init.d/opsware-sas start telldaemon
```

- 5 Optionally check the status of the SA-uCMDB Connector with the following command:

```
/etc/init.d/opsware-sas status telldaemon
```


Editing the Mapping File

All mappings are defined in the mapping.xml configuration file, so administrators can easily view and edit them. The XML mapping file can be modified to change the data being transferred by the SA-uCMDB connector. The mapping file also provides the ability to choose to omit specific CI and attributes.

Permissions: In order to view or edit the mapping.xml file, you must first log in to the SA Core as **root** in order to have read/write privileges.

NOTE: This section describes your editing options within the mapping file. For instructions on the process for customizing the mapping file, including when you need to stop and start the connector in order to make the changes take effect, see [Customizing the Mapping File](#).

Illustration of Mapping File

Here is a snippet of the out-of-the-box mapping file:

```
<Model-Definition model-name='hosts'>
  <CI ucmdb-ci-type-name='node' enable='true' base-class='node'>
    <Attribute source='Node/Name' target-attr='name' enable='true' />
    <Attribute source='Node/Description' target-attr='description'
      enable='true' />
  </CI>
</Model-Definition>
```

where the highlighted text indicates editable fields.

NOTE: See appendix for complete out-of-the-box mapping file.

Each Model Definition tag in the mapping file defines a specific model name. In this example, this Model-Definition defines the 'hosts' model.

Each model can contain many Configuration Items (CIs). Each CI tag defines the composition of the CI. In this example, 'node' is the CI being defined.

For each attribute, **source** indicates the default attribute name in the source database.

- The **target-attr** field specifies the uCMDB attribute name that the source is mapping to.
- The **enable** field defines whether to map the attribute. The default value for **enable** is 'true'; which means the attribute will be loaded into the uCMDB. When you set **enable** to 'false', you are choosing *not* to map the attribute; which means the attribute will *not* be loaded to uCMDB.

XML Attribute Values

The following are the XML attribute values, indicating the editable and non-editable values:

WARNING: Do not change non-editable attribute values. It is crucial that the non-editable values, such as, `source='Node/Name'`, remain unchanged. Changing these values can prevent the synchronization from running properly and can lead to errors.

XML Attribute Tag	Attributes	Sample Attribute Values and Notes	Editable?
Model-Definition	model-name	'hosts', 'sa', 'software', 'compliance', 'hypervisor', 'vmrelations', 'compliance_status'	NOT Editable
	enable	'true' to enable this attribute; 'false' to disable	Editable
CI	ucmdb-ci-type-name	Specifies the uCMDB CI type. For example: 'node', 'ip_address'	Editable
	enable	'true' to enable this attribute; 'false' to disable	Editable
Attribute	source	Specifies the SA custom attribute name. For example: 'Node/Name', 'Node/Description', 'Node/BiosAssetTag', 'Node/BiosSerialNumber', 'Node/Facility', 'Node/VirtualizationTypeId' WARNING: Do not edit the Source value. Modifying the Source value will damage the mapping and may cause errors.	NOT editable
	target-attr	Specifies the uCMDB attribute name that the source is mapping to. For example: 'name', 'description' NOTE: target-attr value must be a unique name.	Editable
	enable	'true' to enable this attribute; 'false' to disable	Editable
	conversion-name	Only used for conversion functions. See Customized conversion function for details. For example: 'com.hp.tell.ConversionMethod\$com.hp.tell.MyConvertVirtualizationType'	Editable
Attribute-Custom	sa-custom-attribute-key-value	Specifies the SA custom attribute name. For example: 'HW_RACK', 'DEVICE_RACK' NOTE: See Support for SA Custom Attributes .	Editable
	target-attr	Specifies the uCMDB attribute name that the source is mapping to. For example: 'serial_number', 'facility' NOTE: target-attr value must be a unique name.	Editable
	enable	'true' to enable this attribute;	Editable

XML Attribute Tag	Attributes	Sample Attribute Values and Notes	Editable?
		'false' to disable	
CI-Filter	enable	'true' to enable this attribute, 'false' to disable NOTE: See Filter Support for modifying CDATA block.	Editable
Relation	ucmdb-relation-type-name	Specifies uCMDB relationship between the CI's. For example: 'containment', 'aggregation'	Editable
	ucmdb-relation-from-ci-type-name	Specifies uCMDB relationship between the CI's of the 'from' CI. For example, if specifying a containment relationship from <i>node</i> to <i>ip_address</i> , the 'node' would be the 'from' CI in this relationship.	Editable
	ucmdb-relation-to-ci-type-name	Specifies the uCMDB relationship between the CI's of the 'to' CI. For example, if specifying a containment relationship from <i>node</i> to <i>ip_address</i> , the 'ip_address' would be the 'to' CI in this relationship.	Editable
	enable	'true' to enable this relationship 'false' to disable	Editable
	ucmdb-relation-id-link	'true' if the relationship contains an ID link. This 'true' value requires the 'from' CI to exist. 'false' if the relationship does not contain an ID link	Editable

Model Definitions

There are 6 models defined in the mapping file that define how data objects are represented in uCMDB. For example, the SA model would represent SA in uCMDB.

Model Definition model-name:	Description
'sa'	generates installed_software.xml
'hosts'	generates node.xml
'software'	generates installed_software.xml
'compliance'	generates policy.xml
'hypervisor'	generates hypervisor.xml
'vmrelations'	generates hypervisorRelation.xml
'compliance_status'	generates policyResult.xml

NOTE: These XML files are generated internally based on the mapping file and should not be edited directly. Editing the generated XML files directly is not supported. Any changes made to the generated files will be overwritten.

Support for SA Custom Attributes

How to transfer SA Custom Attributes to uCMDB

Custom attributes can also be loaded to uCMDB.

In addition to the SA attributes that are synchronized with uCMDB, the out-of-the-box mappings in the new mapping.xml file enables you to specify any SA Custom Attributes defined with an SA Device or inherited from SA Facilities.

Custom Attributes can be specified in the mapping.xml file as follows:

The following example shows how a user could configure the mapping file to extract the custom attribute, *DEVICE_RACK*, and load it to the *my_location_rack* destination in uCMDB. The **enable** attribute is set to 'true', showing that the user chose to load this attribute to uCMDB.

```
<CI ucmdb-ci-type-name='node' enable='true' base-class='node'>
  <Attribute-Custom sa-custom-attribute-key-value='DEVICE_RACK' target-
  attr='my_location_rack' enable='true' />
</CI>
```

where the highlighted text indicates editable fields.

Filter Support for Queries

The mapping.xml file provides the capability to filter specific criteria.

To filter by specific criteria:

- Embed the filtering clause in the CDATA section under CI-Filter tag.
- Specify whether the filter is enabled by supplying the value for **enable** attribute ('true' to enable, 'false' to disable).

NOTE: The CI-Filter specification is based on the SA database and requires knowledge of the SA schema. You can only supply one CI-Filter per CI type. If multiple filters are needed, you can specify a simple filter expression using AND and OR clauses.

Example of a single filter (out-of-the-box mapping):

```
<CI ucmdb-ci-type-name='node' enable='true' base-class='node'>
  <Attribute source='Node/Name' target-attr='name' enable='true' />
  <CI-Filter enable='true'><![CDATA[(DEVICES.OPSW_LIFECYCLE = 'MANAGED')]]></CI-
  Filter>
</CI>
```

In the above example, the filter selects SA devices with State: 'managed'. By default the SA-uCMDB Connector only synchronizes Managed device objects.

Example of a filter which includes an AND clause (modified mapping):

```
<CI-Filter enable='true'><![CDATA[(DEVICES.DVC_MODEL = 'POWEREDGE 2950') and
(DEVICES.DVC_ID > 300000000)]]></CI-Filter>
```

In the above example, the filter selects SA devices with BOTH the Model, 'POWEREDGE 2950', and the ID greater than '300000000'.

Extended Out-Of-The-Box Mappings

The mapping file is provided to enable you to:

- change names of attributes being populated in uCMDB,
- change how data is populated in uCMDB, and
- specify which uCMDB CI type gets populated.

Additional Out-of-the-Box Mappings

The **Facility** and **VirtualizationType** attributes are disabled by default in the out-of-the-box mapping file. However, they may be enabled, as shown below:

ServerVO.getFacility()

```
<Attribute source='Node/Facility' target-attr='facility' enable='true'/>
```

ServerVO.getVirtualizationType()

```
<Attribute source='Node/VirtualizationTypeId' target-attr='virtualization_type_id' enable='true'/>
```

Customized Data Conversion Function

If data to be populated in uCMDB needs to be tailored during synchronization, *custom conversion methods* can be written and provided to the SA-uCMDB Connector. The SA-uCMDB Connector can, then, apply these functions to transform the data from the SA syntax to the desired uCMDB syntax. For example, you can write *custom conversion methods* to convert lower case to upper case, or bytes to megabytes, and so on.

Customized conversion functions should be provided to the SA-uCMDB Connector via a jar file named **tell_conversions.jar**, and placed in **/etc/opt/opsware/tell/lib** prior to the connector startup. After you restart the connector, the custom conversion java class should extend the **ConversionMethod** class and import the **com.hp.tell.ConversionMethod** package.

To customize data conversion:

- 1 If the uCMDB Connector running, you must stop and disable the Connector before editing the mapping file.

- Run the **stop** command to stop the SA-uCMDB Connector:

```
/etc/init.d/opsware-sas stop telldaemon
```

- Run the **disable** command to disable the SA-uCMDB Connector:

```
disable
```

IMPORTANT: Make sure the connector is stopped and disabled. If the connector is not stopped and disabled when you edit the mapping file, you may encounter problems when you try to restart the Connector.

- 2 Write the customized conversation function code in java.

For example, see [Sample Conversion File](#) as an example. In this example, the conversion file's name is: **MyConvertVirtualizationType.java**.

- 3 Modify the mapping.xml file to utilize the conversion file that you just created.

For example, you would place the following line in the mapping.xml file to point to the java file, **MyConvertVirtualizationType.java**:

Original text in mapping file

```
<Attribute source='Node/VirtualizationTypeId' target-attr='virtualization_type_id'  
enable='false' />
```

Customized text in mapping file

```
<Attribute source='Node/VirtualizationTypeId' target-attr='device_isVirtual'  
enable='true' conversion-  
name='com.hp.tell.ConversionMethod$com.hp.tell.MyConvertVirtualizationType' />
```

This modified line of XML has the following values:

- `'device_isVirtual'` is the new attribute value for **target-attr**. Because this conversion changes the data type, it should be mapped to a different uCMDB attribute. However, if you are not changing the data type, then you should map to the same **target-attr** value.*
- `conversion-name` is the XML name for the conversion attribute. This is a verbatim label and cannot be substituted.
- `'com.hp.tell.ConversionMethod$com.hp.tell.MyConvertVirtualizationType'` is the attribute value for conversion-name, and `MyConvertVirtualizationType.java` is the java conversion code file name.

The target-attr value is critical to the success of the conversion operation:

Changing data types

If the conversion is changing an attribute's data type, make sure that the destination attribute (specified by **target-attr**) has the same or compatible requirements, such as length and format. In the previous example, we modified the **target-attr** value because the conversion changes the actual data type. If, for example, you were merely converting the unit of measure (UOM), then you could specify the same **target-attr** value, because the actual data type did not change.

Unique filename per target-attr

Each **target-attr** conversion requires a unique java conversion code filename. The java conversion file represents a singular **target-attr** (output). For example, you can have multiple **target-attr** conversion scenarios for a single source attribute; however, each **target-attr** must be stated on an individual attribute tag in the mapping file, as shown in the following example:

```
<Attribute source='Node/VirtualizationTypeId' target-attr='virtualization_type_id1'  
enable='true' conversion-  
name='com.hp.tell.ConversionMethod$com.hp.tell.MyConvertVirtualizationType1' />  
<Attribute source='Node/VirtualizationTypeId' target-attr='virtualization_type_id2'  
enable='true' conversion-  
name='com.hp.tell.ConversionMethod$com.hp.tell.MyConvertVirtualizationType2' />
```

- 4 Compile the customized conversion file (**MyConvertVirtualizationType.java** in this example). This generates the executable binaries.
- 5 Compress all of the conversion binaries into a jar file with the following name: **tell_conversions.jar**.
NOTE: You must use this exact jar filename for the SA-uCMDB Connector to recognize it.
- 6 Place the jar file in the SA Core directory, **/etc/opt/opsware/tell/lib**, prior to the starting up the uCMDB Connector.
NOTE: You must use this exact directory path for the SA-uCMDB Connector to recognize it.
- 7 Start the uCMDB Connector.

The conversion function will convert the data dynamically, at the time the SA-uCMDB Connector is restarted.

Sample Conversion File – MyConvertVirtualizationType.java

This sample conversion file provides sample java code to use as a guideline. This java sample converts an SA **VirtualizationType** from Type: *Numeric* into Type: *String* for uCMDB.

NOTE: You can only have one attribute conversion per java file. To convert multiple attributes, you need to have multiple java files. Each target attribute can only have one conversion.

Tip: Name the conversion file based on the attribute being modified. As in this example, the java filename is **MyConvertVirtualizationType** because it is modifying the **VirtualizationType** attribute.

```
package com.hp.tell;

import java.math.BigDecimal;

import com.hp.tell.ConversionMethod;

public class MyConvertVirtualizationType extends ConversionMethod {

    public Object convert(Object value) throws Exception{

        Integer vType = putInteger(value);
        String vValue;

        /*
        * Function to convert SA VirtualizationType (numeric) to string type
        For uCMDB.
        */

        if (vType > 0) {
            vValue = "True";
        } else {
            vValue = "False";
        }

        return vValue;
    }

    private Integer putInteger(Object o) throws Exception {
        if (o instanceof String) {
            return Integer.valueOf((String) o);
        }
        if (o instanceof BigDecimal) {
            return ((BigDecimal)o).intValue();
        }
        if (o instanceof Integer) {
            return (Integer)o;
        }

        throw new Exception("Invalid conversion in putInteger
"+o.getClass().toString());
    }
}
```

Managing the SA-uCMDB Connector

Stopping and Disabling the SA-uCMDB Connector

If the SA-uCMDB Connector is running, you must stop and disable the Connector before making any kind of configuration change.

To stop and disable the SA-uCMDB Connector:

- 1 Run the **stop** command to stop the SA-uCMDB Connector:

```
/etc/init.d/opsware-sas stop telldaemon
```

- 2 Run the **disable** command to disable the SA-uCMDB Connector:

```
disable
```

IMPORTANT: Make sure the connector is stopped and disabled before making any configuration change. If the connector is not stopped and disabled, you may encounter problems when you try to restart the Connector.

The stop Command

When you stop the SA-uCMDB Connector, it will stop transferring data from the SA database to uCMDB. To stop the SA-uCMDB Connector, enter the following command on an SA core server:

```
/etc/init.d/opsware-sas stop telldaemon
```

This stops the SA-uCMDB Connector.

If the SA-uCMDB Connector is disabled, the output will look like the following:

```
opsware-sas: One or more of the specified components does not exist
in the following file:
/opt/opsware/oi_util/startup/components.config
```

If you no longer need the SA-uCMDB Connector, you can disable it with the `disable` command. For more information, see [The disable Command](#).

The disable Command

Use the **disable** command to disable the SA-uCMDB Connector. If the SA-uCMDB Connector is running, the **disable** command will stop it before disabling it. If the SA-uCMDB Connector is disabled, you will not be able to start it.

The **disable** command modifies the file

/opt/opsware/oi_util/startup/components.config and comments out the lines for the **telldaemon**, which is the process for the SA-uCMDB Connector.

Location of the disable Command

The **disable** command is located on your SA core server in the directory **/opt/opsware/tell/bin**.

Syntax of the disable Command

```
disable
```


Enabling and Starting the SA-uCMDB Connector

Before starting the SA-uCMDB Connector, you must enable it to make sure the most up to date configuration elements are engaged.

To enable and start the SA-uCMDB Connector:

- 1 Run the **enable** command to change the configurations of the SA-uCMDB Connector. There are multiple options for the enable command depending on your configuration.

The following is a simple example of this command:

```
enable --host myserver01.hp.com --port 8888 --user ucmdb-admin
--password leM93A3dme
```

For more information about the complete set of parameters, syntax, and options, see [The enable Command](#).

- 2 Run the **start** command to restart the uCMDB Connector:

```
/etc/init.d/opsware-sas start telldaemon
```

- 3 Optionally check the status of the SA-uCMDB Connector with the following command:

```
/etc/init.d/opsware-sas status telldaemon
```

For more information, see [Displaying the Status of the SA-uCMDB Connector](#).

The enable Command

Before you can start the SA-uCMDB Connector, you must enable it with the enable command. When you enable it, you provide the uCMDB server name or IP address, port number, login, and password.

Use the **enable** command to configure and enable the SA-uCMDB Connector. This section describes the **enable** command. You must enable the SA-uCMDB Connector before you can start it.

The **enable** command does the following:

- Modifies the SA-uCMDB Connector configuration file **/etc/opt/opsware/tell/tell.conf**. and enters the uCMDB server's host name or IP address, port number, and login into this file.
- Saves the user's password.
- Modifies the file **/opt/opsware/oi_util/startup/components.config** and uncomments the lines for the **telldaemon**, which is the process for the SA-uCMDB Connector.

If you modify any of the uCMDB configuration parameters while the SA-uCMDB Connector is running, you must stop and restart the SA-uCMDB Connector for your changes to take effect.

Location of the enable Command

The **enable** command is located on your SA core server in the directory

```
/opt/opsware/tell/bin.
```

New Syntax in the enable Command

In SA 9.14, additional parameters were added to the SA-uCMDB Connector's **enable** command in order to support the new uCMDB Browser. The new parameters are described in this section.

```
enable [--protocol <ucmdb_protocol>] [--host <ucmdb_host_ip>] [--port
<ucmdb_host_port_number>] [--browser_protocol <ucmdb_browser_protocol>] [--
browser_host <ucmdb_browser_host_ip>] [--browser_port <ucmdb_browser_host_port>] [--
user <ucmdb_admin_user>] [--password <ucmdb_admin_password>] [--help]
```

Parameter	Description	New
<code>--protocol</code> <code><ucmdb_protocol></code>	uCMDB server protocol, http or https. Default is http.	New
<code>--host</code> <code><ucmdb_host_ip></code>	This option gives the IP address or host name of your HP uCMDB server. The default value is localhost.	
<code>--port</code> <code><ucmdb_host_port_number></code>	This option gives the port number of your HP uCMDB server. The default value is 8080.	
<code>--browser_protocol</code> <code><ucmdb_browser_protocol></code>	uCMDB Browser server protocol, http or https. Default is http.	New
<code>--browser_host</code> <code><ucmdb_browser_host_ip></code>	This option gives the IP address or host name of your HP uCMDB Browser host name or IP. The default value is localhost.	New
<code>--browser_port</code> <code><ucmdb_browser_host_port></code>	This option gives uCMDB Browser host port. The default value is 8080	New
<code>--user</code> <code><ucmdb_admin_user></code>	This option gives the user name of an administrative user for your HP uCMDB server. The default value is admin.	
<code>--password</code> <code><ucmdb_admin_password></code>	This option gives the password for the user provided in the <code>--user</code> option. The default value is admin.	

Example of **enable** command without SSL enabled:

```
enable --protocol http --host 192.168.8.93 --port 9999 --browser_protocol http --
browser_host 192.168.8.100 --browser_port 8888 --user john-ucmdb --password
mypass1234
```

Example of **enable** command with SSL enabled for the uCMDB Server and the uCMDB Browser:

```
enable --protocol https --host 192.168.8.93 --port 9999 --browser_protocol https --
browser_host 192.168.8.100 --browser_port 8888 --user john-ucmdb --password
mypass1234
```

Displaying the Status of the SA-uCMDB Connector

To display the status of the SA-uCMDB Connector, enter the following command on an SA core server:

```
/etc/init.d/opsware-sas status telldaemon
```

If the SA-uCMDB Connector is enabled but not running, the output will look like the following:

```
Verify "telldaemon" running: FAILURE (pidfile does not exist)
Failed to perform "status" operation on Opsware SAS components.
```

If the SA-uCMDB Connector is disabled, the output will look like the following:

```
opsware-sas: One or more of the specified components does not exist
in the following file:
/opt/opsware/oi_util/startup/components.config
```

SA-uCMDB Data Relationship and Transfer

CI Relationships Maintained

The following table lists the Configuration Item (CI) relationships maintained by the SA-uCMDB Connector.

CI Relationships Maintained

From uCMDB CI	Via	To uCMDB CI
Node	containment	IpAddress
Node	composition	InstalledSoftware
Node	composition	Hypervisor
Node	aggregation	PolicyResult
Hypervisor	ExecutionEnvironment	Node
Policy	composition	PolicyResult
SASystem	aggregation	Node
SASystem	aggregation	Policy

Example uCMDB Showing an SA Managed Server

Figure 1 is from an HP uCMDB screen and it shows:

- One SA managed server named “atari”.
- The managed server’s IP address 10.12.10.65.
- The managed server “atari” is running a VMware hypervisor.
- Two virtual machines are running on the hypervisor named “gofish” and “miracles”.

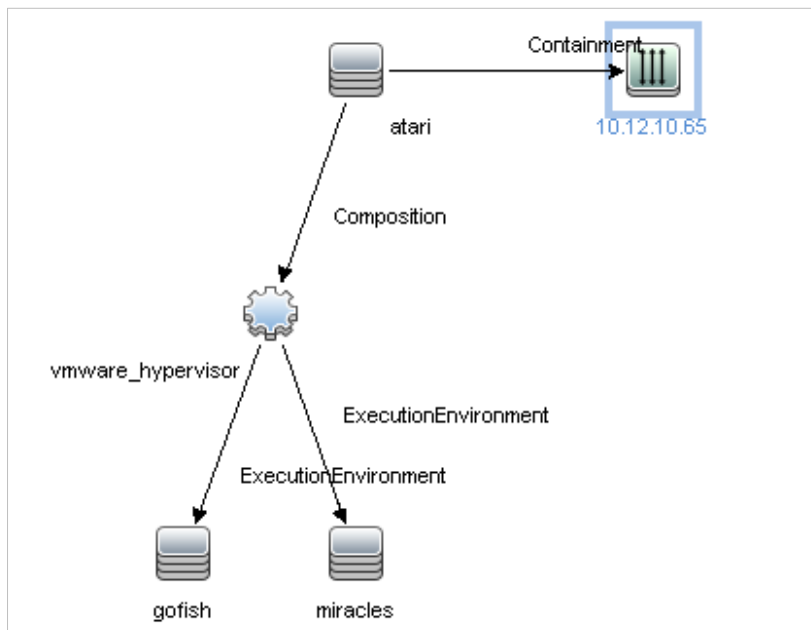


Figure 1 SA Managed Servers Displayed in the uCMDB

SA Data Transferred to uCMDB

The following data from the SA database is transferred to the uCMDB Configuration Items (CI) and Attributes:

uCMDB CIs and Attributes Populated by SA

uCMDB CI	uCMDB Attribute
Node	Name
Node	Description
Node	BiosAssetTag
Node	DefaultGatewayIpAddress
Node	NodeModel
Node	SerialNumber
Node	BiosUuid
Node	NetBiosName
Node	MemorySize
Node	OsDescription
Node	OsFamily
Node	TenantOwner
IpAddress	Name
IpAddress	RoutingDomain
InstalledSoftware	Name
InstalledSoftware	Vendor
InstalledSoftware	BuildNumber
InstalledSoftware	DmlProductName
Hypervisor	Name
Hypervisor	Description
Hypervisor	ProductName
Policy	Name
Policy	Description
Policy	PolicyCategory
Policy	PolicyDefinedBy
PolicyResult	Name

uCMDB CI	uCMDB Attribute
PolicyResult	PolicyResultDateTime
PolicyResult	ComplianceStatus
PolicyResult	RulesCompliant
PolicyResult	RulesNonCompliant
PolicyResult	ComplianceLevel
SASystem	Name
SASystem	Description
SASystem	Version

Frequency of Data Transfer to uCMDB

When the SA-uCMDB Connector first starts running, it queries the SA database, creates the CIs in the uCMDB and transfers the data from SA to the uCMDB. After that, whenever the data in the SA database changes, the SA-uCMDB Connector automatically detects the changes and transfers the modified data to the uCMDB. The connector logs information in the log file `/var/log/opsware/tell/LOAD_STATS.0.log`.

For the complete list of data transferred from SA to the uCMDB, see [SA Data Transferred to uCMDB](#).

Accessing the uCMDB Browser from the SA Client

The uCMDB Browser Window

You can view server details in the uCMDB Browser window.

To view server details:

- 1 Log in to the SA Client.
- 2 Go to **Devices > All Managed Servers**.
- 3 Select any server and click **Actions > Open with uCMDB Browser**.

Optional: You can also use the context menu here or on the search panel. Select the server, then right-click and choose **Open With > uCMDB browser**.

Sample URL that SA uses to open the uCMDB Browser for a specific Managed Server:

```
http://my-ucmdb.mycomp.com:8080/ucmdb-api/ucmdb-browser/?locale=en&theme=LIGHT#refocus-selection=<global_ucmdb_id>
```

- 4 If you are not already logged in to the uCMDB Browser, this will invoke the uCMDB Browser Login screen. Complete using your uCMDB login credentials. You will only need to sign in once per session.

Tip: If a blank page or a Page Not Found error occurs when you open the uCMDB Browser, it could mean that either uCMDB is not set up or the uCMDB server is not running or configured correctly. Make sure that the uCMDB server is configured and that the Tellconnector is running.

If the SA-uCMDB Connector has not been configured and you need to disable the 'Open with uCMDB Browser' menu item, go to **System Configuration > Opsware > Tell** and set the values to no value for **uCMDB Browser URL** and **uCMDB URL**.

Configuring the uCMDB Browser

If the uCMDB Browser needs to be invoked from the SA Client, the uCMDB Browser's related parameters need to be specified after enabling the SA-uCMDB Connector via the following

/opt/opsware/tell/bin/enable parameters:

```
--browser_protocol    - uCMDB Browser server protocol, http or https
--browser_host        - uCMDB Browser host name or IP
--browser_port        - uCMDB Browser host port
```

Also, by default the SA Client invokes the uCMDB Browser via the uCMDB 9.05-compatible URL prefix.

For example, to use the uCMDB 10.01-based Browser:

- 1 Stop the SA-uCMDB Connector by running the **stop** command:

```
/etc/init.d/opsware-sas stop telldaemon
```

- 2 Disable the SA-uCMDB Connector by running the **disable** command:

```
disable
```

IMPORTANT: Make sure the connector is stopped and disabled. If the connector is not stopped and disabled when you revise the configuration file, you may encounter problems when you try to restart the Connector.

- 3 Update the uCMDB Browser prefix in the SA-uCMDB Connector configuration file **/etc/opt/opsware/tell/tell.conf** to indicate the correct uCMDB version.

For example:

Change from the following uCMDB 9.05 default:

```
com.hp.sa.tell.ucmdb.browser.path.suffix=/ucmdb-api/ucmdb-browser
```

To the uCMDB 10.01 prefix:

```
com.hp.sa.tell.ucmdb.browser.path.suffix=/ucmdb-browser
```

- 4 After the configuration file is updated, enable the SA-uCMDB Connector by running the **enable** command.

NOTE: The syntax of the enable command varies depending on your environment. See [The enable Command](#) in this document for an explanation of the enable command syntax and options.

- 5 Restart the uCMDB Connector. Enter the following command to start the SA-uCMDB Connector:

```
/etc/init.d/opsware-sas start telldaemon
```

- 6 Optionally check the status of the SA-uCMDB Connector with the following command:

```
/etc/init.d/opsware-sas status telldaemon
```

Support for uCMDB Server Versions 9.05 and 10.01

For the SA-uCMDB Integration, SA 9.14 or later supports the following uCMDB Server integrations:

- uCMDB 9.05 Content Pack 10 or higher, Cumulative Update Package 6 or higher
- uCMDB 10.01 Content Pack 12

NOTE: Version support and compatibility information is subject to change. For complete and up-to-date support and compatibility information, see the support matrix for the relevant product release. All support matrices and product manuals are available here on the HP Software Support Online web site:

http://support.openview.hp.com/sc/support_matrices.jsp

You can also download the HP Server Automation Support and Compatibility Matrix for this release from the HP Software Support Online Product Manuals web site:

<http://support.openview.hp.com/selfsolve/manuals>

Global uCMDB IDs

With uCMDB 9.04 and earlier, only the local uCMDB ID's as known to that uCMDB server were synchronized in SA.

With uCMDB 9.05 and later, the uCMDB Servers can be configured as uCMDB Global ID generators, where the uCMDB ID's generated are *global* and *unique* in multi-uCMDB server environments. In such environments, these global ID's are needed to properly invoke the uCMDB Browser.

The SA 9.14 SA-uCMDB Connector was enhanced to automatically use the global uCMDB ID of CI's if the uCMDB Server is configured as a global ID generator. No special configuration is needed for the SA-uCMDB Connector.

SSL Connectivity to the uCMDB Server and the uCMDB Browser

The SA-uCMDB Connector supports SSL protocol for the uCMDB Server and the uCMDB Browser.

When enabling Secure Sockets Layer Communication (SSL), the appropriate certificate and keystore need to be in place for the SA-uCMDB Connector.

To enable SSL:

- 1 Follow the instructions in the *uCMDB Deployment Guide*, "Enabling Secure Sockets Layer Communication," to create a uCMDB keystore and export the certificate to a file.
- 2 Import exported certificate from step 1 to where the SA-uCMDB Connector is installed. For example, the keystore must be placed in **/var/opt/opsware/crypto/tell** with the keystore filename: **tell.keystore** and the keystore password: **hppass**.

Example of import command:

```
/opt/opsware/jdk1.6/bin/keytool -import -noprompt -alias hpsaucmdb -file  
<path_to_the_exported_hpcert> -keypass hppass -keystore  
/var/opt/opsware/crypto/tell/tell.keystore -storepass hppass
```


Troubleshooting Tips

Running the SA-uCMDB Connector on a Second Core

In some cases, a particular core in a multi-master SA Mesh needs to be deactivated and it becomes necessary to run the SA-uCMDB Connector from a different core in that mesh. Sometimes this is also needed if network performance from another core to the uCMDB server is preferred. In those scenarios, the following steps are necessary:

To run the connector on a second core:

- 1 Stop the SA-uCMDB Connector on the first core and remove its affinity to it.

```
/etc/init.d/opsware-sas stop telldaemon  
/opt/opsware/tell/bin/tell --release
```

- 2 On the second core, enable the SA-uCMDB Connector by running the **enable** command.

NOTE: The syntax of the enable command varies depending on your environment. See [The enable Command](#) in this document for an explanation of the enable command syntax and options.

- 3 Take responsibility of the SA-uCMDB integration, and then start the SA-uCMDB Connector.

```
/opt/opsware/tell/bin/tell --take  
/etc/init.d/opsware-sas start telldaemon
```

To enable additional logging:

- 1 Start the SA-uCMDB Connector.

Normal log files are stored in the **/var/log/opsware/tell** directory. Default file names include the following:

```
tell.0.log           (normal startup log)  
ucmdb_failure.*.log (ucMDB failures seen during synchronization)  
LOAD_STATS.*.log   (number of processed data)
```

- 2 To request additional logging details, specify the requested information in the **/etc/opt/opsware/tell/logging.properties** file as follows:

Field	Description
<code>java.util.logging.FileHandler.limit</code>	Specifies the maximum number of bytes to write to any one file, default value is 10000000
<code>java.util.logging.FileHandler.count</code>	Specifies the number of files to use, default value is 10
<code>java.util.logging.FileHandler.append</code>	Specifies append mode, default to true
<code>java.util.logging.FileHandler.pattern</code>	Specifies the pattern for naming the output file where the log file can be found, default to /var/log/opsware/tell/tell.%g.log

CAUTION: Use caution when modifying the file limit. Large numbers might impact performance.

On-demand Synchronization

Upon SA restart, the SA-uCMDB Connector normally continues synchronizing SA data to uCMDB from where it ended before the restart. The connector also runs a full sync, periodically. However, in some cases, such as when there are networking or server issues that prevent the updates from reaching the uCMDB server, you may need to trigger the full sync on demand.

To trigger the synchronization on demand:

- 1 Stop the SA-uCMDB Connector.
- 2 Restart the SA-uCMDB Connector with the following option:

```
/opt/opsware/tell/bin/tell --startfresh
```

Viewing Log Files

The SA-uCMDB Connector generates the following text log files. You can view these log files in a text editor to get more information.

- **/var/log/opsware/tell/tell.0.log** is the main log file for information, warnings and errors encountered by the SA-uCMDB Connector.
- **/var/log/opsware/tell/LOAD_STATS.0.log** contains the status and statistics of the initial data load, and approximate times to complete the initial data load.
- **/var/log/opsware/tell/ucmdb_failure.0.log** contains uCMDB errors, primarily reconciliation errors if the SA data is incomplete, for example, if the required uCMDB keys are missing. This could happen if a server did not have a serial number or an IP address, for example. This log file contains the uCMDB exception, the reason why it failed and a trace of the CIs that caused the exception.

SA-uCMDB Connector Daemon

The SA-uCMDB Connector runs the daemon **/etc/opt/opsware/startup/telldaemon** on your SA core server. Make sure this process is running on your SA core server.

If it is not running, start it as described in [Enabling and Starting the SA-uCMDB Connector](#).

If it is running, check the status as described in [Displaying the Status of the SA-uCMDB Connector](#)

Appendix A – Mapping File

```
<?xml version='1.0' ?>
<DB-UCMDB-HIGHLEVEL-MAPPING>
  <!-- generates installed software.xml -->
  <Model-Definition model-name='sa' enable='true'>
    <CI ucmdb-ci-type-name='server_automation_system' enable='true' base-
class='server_automation_system'>
      <Attribute source='SA/Description' target-attr='description'
enable='true' />
      <Attribute source='SA/Name' target-attr='name' enable='true' />
      <Attribute-Default target-attr='version' target-attr-value='9.14'
enable='true' />
    </CI>
  </Model-Definition>

  <!-- generates node.xml -->
  <Model-Definition model-name='hosts' enable='true'>
    <CI ucmdb-ci-type-name='server_automation_system' reference-ci='true'
enable='true' />

    <CI ucmdb-ci-type-name='ip_address' enable='true' base-class='node'>
      <Attribute source='IpAddress/PrimaryIpAddress' target-attr='name'
enable='true' />
      <Attribute source='IpAddress/RoutingDomain' target-attr='routing_domain'
enable='true' />
    </CI>

    <CI ucmdb-ci-type-name='node' enable='true' base-class='node'>
      <Attribute source='Node/Name' target-attr='name' enable='true' />
      <Attribute source='Node/Description' target-attr='description'
enable='true' />
      <Attribute source='Node/BiosAssetTag' target-attr='bios_asset_tag'
enable='true' />
      <Attribute source='Node/BiosSerialNumber' target-attr='serial_number'
enable='true' />
      <Attribute source='Node/BiosUuid' target-attr='bios_uuid'
enable='true' />
      <Attribute source='Node/DefaultGatewayIpAddress' target-
attr='default_gateway_ip_address' enable='true' />
      <Attribute source='Node/NetBiosName' target-attr='net_bios_name'
enable='true' />
      <Attribute source='Node/NodeModel' target-attr='node_model'
enable='true' />
      <Attribute source='Node/MemorySize' target-attr='memory_size'
enable='true' />
      <Attribute source='Node/OsDescription' target-attr='os_description'
enable='true' />
      <Attribute source='Node/OsFamily' target-attr='os_family'
enable='true' />
      <Attribute source='Node/TenantOwner' target-attr='TenantOwner'
enable='true' />
      <Attribute source='Node/Facility' target-attr='facility'
enable='false' />
      <Attribute source='Node/VirtualizationTypeId' target-
attr='virtualization_type_id' enable='false' />
      <Attribute source='IpAddress/ManagementIpAddress' target-attr='ip_address'
enable='false' />
      <CI-Filter enable='true'><![CDATA[(DEVICES.OPSW_LIFECYCLE =
'MANAGED')]]></CI-Filter>
    </CI>

    <Relation ucmdb-relation-type-name='containment' ucmdb-relation-from-ci-
type-name='node' ucmdb-relation-to-ci-type-name='ip_address' enable='true' ucmdb-
relation-id-link='true' />
  </Model-Definition>
</DB-UCMDB-HIGHLEVEL-MAPPING>
```

```

        <Relation ucmdb-relation-type-name='aggregation' ucmdb-relation-from-ci-
type-name='server_automation_system' ucmdb-relation-to-ci-type-name='node'
enable='true' ucmdb-relation-id-link='false'/>
    </Model-Definition>

    <!-- generates installed_software.xml -->
    <Model-Definition model-name='software' enable='true'>
        <CI ucmdb-ci-type-name='node' base-class='node' reference-ci='true'
enable='true'/>

        <CI ucmdb-ci-type-name='installed_software' enable='true' base-
class='installed_software'>
            <Attribute source='InstalledSoftware/DmlProductName' target-
attr='dml_product_name' enable='true'/>
            <Attribute source='InstalledSoftware/Name' target-attr='name'
enable='true'/>
            <Attribute source='InstalledSoftware/Version' target-attr='version'
enable='true'/>
            <Attribute source='InstalledSoftware/Vendor' target-attr='vendor'
enable='true'/>
        </CI>

        <Relation ucmdb-relation-type-name='composition' ucmdb-relation-from-ci-
type-name='node' ucmdb-relation-to-ci-type-name='installed_software' ucmdb-relation-
id-link='true' enable='true'/>
    </Model-Definition>

    <!-- generates policy.xml -->
    <Model-Definition model-name='compliance' enable='true'>
        <CI ucmdb-ci-type-name='server_automation_system' reference-ci='true'
enable='true'/>

        <CI ucmdb-ci-type-name='policy' base-class='policy' enable='true'>
            <Attribute source='Policy/Name' target-attr='name' enable='true'/>
            <Attribute source='Policy/Description' target-attr='description'
enable='true'/>
            <Attribute-Default target-attr='policy_defined_by' target-attr-
value='SA' enable='true'/>
            <Attribute-Default target-attr='policy_category' target-attr-
value='audit' enable='true'/>
        </CI>

        <Relation ucmdb-relation-type-name='aggregation' ucmdb-relation-from-ci-
type-name='server_automation_system' ucmdb-relation-to-ci-type-name='policy'
enable='true' ucmdb-relation-id-link='false'/>
    </Model-Definition>

    <!-- generates hypervisor.xml -->
    <Model-Definition model-name='hypervisor' enable='true'>
        <CI ucmdb-ci-type-name='node' base-class='node' reference-ci='true'
enable='true'/>

        <CI ucmdb-ci-type-name='hypervisor' base-class='hypervisor' enable='true'>
            <Attribute source='Hypervisor/Name' target-attr='name' enable='true'/>
            <Attribute source='Hypervisor/Description' target-attr='description'
enable='true'/>
            <Attribute source='Hypervisor/ProductName' target-attr='product_name'
enable='true'/>
        </CI>

        <Relation ucmdb-relation-type-name='composition' ucmdb-relation-from-ci-
type-name='node' ucmdb-relation-to-ci-type-name='hypervisor' ucmdb-relation-id-
link='true' enable='true'/>
    </Model-Definition>

```

```

    <!-- generates hypervisorRelation.xml -->
    <Model-Definition model-name='vmrelations' enable='true'>
      <CI ucmdb-ci-type-name='hypervisor' base-class='hypervisor' reference-
ci='true' enable='true'/>

      <CI ucmdb-ci-type-name='node' base-class='node' reference-ci='true'
enable='true'/>

      <Relation ucmdb-relation-type-name='execution_environment' ucmdb-relation-
from-ci-type-name='hypervisor' ucmdb-relation-to-ci-type-name='node' ucmdb-relation-
id-link='false' enable='true'/>
    </Model-Definition>

    <!-- generates policyResult.xml -->
    <Model-Definition model-name='compliance_status' enable='true'>
      <CI ucmdb-ci-type-name='policy' base-class='policy' reference-ci='true'
enable='true'/>

      <CI ucmdb-ci-type-name='node' base-class='node' reference-ci='true'
enable='true'/>

      <CI ucmdb-ci-type-name='policy_result' base-class='policy_result'
enable='true'>
        <Attribute source='PolicyResult/Name' target-attr='name' enable='true'/>
        <Attribute source='PolicyResult/ComplianceStatus' target-
attr='compliance_status' enable='true'/>
        <Attribute source='PolicyResult/PolicyResultDateTime' target-
attr='policy_result_date_time' enable='true'/>
        <Attribute source='PolicyResult/RulesCompliant' target-
attr='rules_compliant' enable='true'/>
        <Attribute source='PolicyResult/RulesNonCompliant' target-
attr='rules_non_compliant' enable='true'/>
        <Attribute source='PolicyResult/ComplianceLevel' target-
attr='compliance_level' enable='true'/>
      </CI>

      <Relation ucmdb-relation-type-name='composition' ucmdb-relation-from-ci-
type-name='policy' ucmdb-relation-to-ci-type-name='policy_result' ucmdb-relation-id-
link='false' enable='true'/>
      <Relation ucmdb-relation-type-name='aggregation' ucmdb-relation-from-ci-
type-name='node' ucmdb-relation-to-ci-type-name='policy_result' ucmdb-relation-id-
link='true' enable='true'/>
    </Model-Definition>
  </DB-UCMDB-HIGHLEVEL-MAPPING>

```