

HP Universal CMDB

Software Version: 9.05, CP 11.0

Atrium Integration (Revised Edition)

Document Release Date: October 2013

Software Release Date: June 2012



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 Notice

© Copyright 2002 - 2013 Hewlett-Packard Development Company, L.P.

Trademark Notices

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

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

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

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

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

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

UNIX® is a registered trademark of The Open Group.

Acknowledgements

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

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

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

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

<http://www.hp.com/go/hpsoftwaresupport>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

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

To find more information about access levels, go to:

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

Atrium Integration

This chapter includes:

Overview	7
Supported Versions	7
How to Work with the Data Push into Atrium Adapter	7
How to Work with the Population from Atrium Adapter	12
Atrium Push Job	15
Import Data from Atrium Job	17
Mapping Files	19
Troubleshooting and Limitations	24

Overview

UCMDB-Atrium integration consists of two independent, bi-directional parts: the **Data Push into Atrium** and the **Population from Atrium**.

- The **Data Push into Atrium** in UCMDB replicates CIs and relationships to Atrium and Remedy.

The out-of-the-box integration does not transfer a specific list of CIs and relationships, but does enable you to replicate any CI or relationship from UCMDB to Remedy or Atrium.

For examples of enabling the integration with commonly used CIs and relationships, see ["Configure synchronization queries" on page 10](#).

- The **Population from Atrium** in UCMDB pulls CIs and relationships from Atrium to UCMDB.

Supported Versions

HP Universal CMDB integrates with the following BMC products:

- BMC Remedy Service Desk (Remedy) versions 7.0, 7.1, 7.5, 7.6
- BMC Atrium CMDB (Atrium) versions 2.0, 2.1, 7.5.x, 7.6.x and earlier, 8.1

How to Work with the Data Push into Atrium Adapter

This task includes the following steps:

- ["Prerequisite- Set up protocol credentials" on the next page](#)
- ["Configure the Properties file" on the next page](#)
- ["Configure the Data Flow Probe" on the next page](#)
- ["Configure synchronization queries" on page 10](#)
- ["Create XML mapping files" on page 10](#)
- ["Create an integration point" on page 11](#)
- ["Define a Job" on page 11](#)
- ["Invoke a full run of the job" on page 11](#)

1. Prerequisite- Set up protocol credentials

Make sure that you have set up the Remedy protocol. For credential information, see "Supported Protocols" in the *HP Universal CMDB Discovery and Integration Content Guide*.

2. Configure the Properties file

Configure the `push.properties` file: **Data Flow Management > Adapter Management > Resources > Packages > AtriumPushAdapter > Configuration Files > push.properties.**

Property	Description
<code>jythonScript.name</code>	The name of the Jython script that is invoked by this push adapter.
<code>mappingFile.default</code>	The default XML mapping file used by mapping if a specific XML mapping file is not defined for an integration query. At least one default mapping file must be present in every adapter.
<code>DebugMode</code>	If this value is set to true , the CI and relationships being pushed to Remedy/Atrium are also saved to XML files on the Data Flow Probe, under the following folder: /discoveryResource/AtriumPushAdapter/work.
<code>smartUpdateIgnoreFields</code>	A comma separated list of attributes (transferred from UCMDB to Atrium) that should not be used to check whether a CI has changed in Atrium. For example, as updateTime always changes, you would not want to update a CI in Atrium just because this attribute has changed.
<code>sortCSVFields</code>	Parameter that includes the TQL results of CSV aggregated fields that must always be sorted. When child attribute values are mapped and aggregated as CSV, the results are not sorted. This can trigger an update, even though nothing has changed in Atrium. To prevent an update, add here the CSV aggregated fields that must always be sorted.
<code>testConnNameSpace</code>	Must be set to the BMC NameSpace being used for test connection purposes (for example, BMC.CORE).
<code>testConnClass</code>	Must be set to the name of a BMC class, to query for connection test purposes (for example, BMC_ComputerSystem).

3. Configure the Data Flow Probe

- a. **For Atrium 7.6.04 and earlier versions:** Copy the JAR and DLL files listed in Table 1 below from the BMC server to the following directory on the Data Flow Probe server:
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumPushAdapter.

For Atrium 8.1: Copy the files **arapi81_build001.jar** and **cmdbapi.jar** from the BMC server to the following directory on the Data Flow Probe server:

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumPushAdapter.

The directory

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumPushAdapter is automatically created once the **AtriumPushAdapter** package is deployed on the UCMDB Server. If it is not present, ensure that the **AtriumPushAdapter** package has been correctly deployed on the UCMDB Server.

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

Table 1

JAR Files	DLL Files
arapi75.jar	arapi75.dll
arutil75.jar	arencrypt75.dll
cmdbapi75.jar	arjni75.dll
commons-beanutils.jar	arrpc75.dll
commons-codec-1.3.jar	arutiljni75.dll
commons-collections-3.2.jar	arutil75.dll
commons-configuration-1.3.jar	arxmlutil75.dll
commons-digester-1.7.jar	cmdbapi75.dll
commons-lang-2.2.jar	cmdbjni75.dll
log4j-1.2.14.jar	icudt32.dll
oncrpc.jar	icuinbmc32.dll
spring.jar	icucbmc32.dll
	Xalan-Cbmc_1_9.dll
	XalanMessagesbmc_1_9.DLL
	xerces-cbmc_2_6.dll
	xerces-depdombmc_2_6.dll

Note:

- The AR System Java API is forward and backward compatible with other versions of the AR System. For a complete compatibility matrix, refer to the "API Compatibility" section in the *BMC Remedy/Atrium Developer Reference Guide*.
- The arencrypt*.dll files are only required if encryption is enabled on the Remedy server.

- b. Edit the **WrapperGateway.conf** file (or **WrapperManager.conf** if the Probe Manager and Gateway are running in separate mode) in the following directory:

C:\hp\UCMDB\DataFlowProbe\bin.

Add the following line after the **wrapper.java.library.path.2=%content_dll%** line:

```
wrapper.java.library.path.3=%runtime%/probeManager
/discoveryResources/AtriumPushAdapter
```

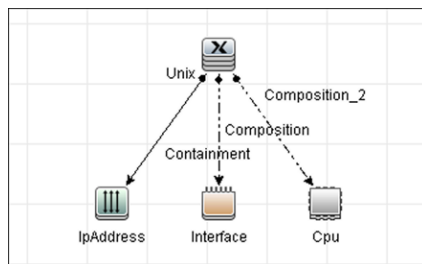
- c. **For Atrium 7.6.04 and earlier versions only:** Add the complete path to the Atrium DLL files (for example, **C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumPushAdapter**) to the Windows System Path on the Data Flow Probe machine.

- d. Restart the Data Flow Probe service.

4. Configure synchronization queries

The CIs and relationships to be pushed to Remedy/Atrium must be queried from UCMDB. Create queries (of type **Integration**) to query the CIs and relationships that have to be pushed to Remedy/Atrium.

An example of such a query (**atrium_push_sample_query**) is included with the Atrium package. To access the query, navigate to **Modeling > Modeling Studio > Root > Integration > Atrium**.



5. Create XML mapping files

For every query created in the step above, create an XML mapping file with the same name as the integration query (the name must have the same case) in the following directory:

C:\hp\UCMDB\UCMDBServer\runtime\fcmdb\CodeBase\AtriumPushAdapter\ mappings

A sample mapping file (**atrium_push_sample_query.xml**) is provided out-of-the-box with the Atrium package.

For more details, see ["Mapping Files" on page 19](#).

6. Create an integration point

For details about creating an integration point, see "Integration Point Pane" in the *HP Universal CMDB Data Flow Management Guide*.

- a. In the Integration Studio, create an integration point, selecting the **Data Push into Atrium** adapter. Enter the following information:

Name	Description
Credentials	<ul style="list-style-type: none">o Select Remedy Protocol.o Select the credentials to be used with this integration point. <p>For credential information, see "Supported Protocols" in the <i>HP Universal CMDB Discovery and Integration Content Guide</i>.</p>
Hostname/IP	The host name or IP address of the BMC Remedy server.
Integration Name	The name you give to the integration point.
Is Integration Activated	Select this check box to create an active integration point. You clear the check box if you want to deactivate an integration, for instance, to set up an integration point without actually connecting to a remote machine.
Port	The port number of the BMC Remedy server.
Probe Name	Select the Probe that should run this integration.


- b. Test the connection. If a connection is not successfully created, check the integration point parameters and try again.
- c. Save the integration point.

7. Define a Job

For details, see "New Integration Job/Edit Integration Job Dialog Box" in the *HP Universal CMDB Data Flow Management Guide*.

Select the queries that will synchronize data between UCMDB and Remedy/Atrium. Save the job definition and the integration point.

8. Invoke a full run of the job

In the Integration Studio, on the Job Definition tool bar, click  to run a full discovery job. For details, see "Integration Jobs Pane" in the *HP Universal CMDB Data Flow Management Guide*.

How to Work with the Population from Atrium Adapter

This task includes the following steps:

1. Prerequisites - File preparation

- a. **For Atrium 7.6.04 and earlier versions:** Locate the files listed in Table 2 on the Remedy ARS and Atrium system and copy them to:

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumImportAdapter

Note: All the files listed in Table 2 are required.

For Atrium 8.1: Locate the files **arapi81_build001.jar** and **cmdbapi.jar** on the Remedy ARS and Atrium system and copy them to:

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumImportAdapter

Table 2

JAR Files	DLL Files
arapi75.jar	arapi75.dll
arutil75.jar	arencrypt75.dll
cmdbapi75.jar	arjni75.dll
commons-beanutils.jar	arrpc75.dll
commons-codec-1.3.jar	arutiljni75.dll
commons-collections-3.2.jar	arutil75.dll
commons-configuration-1.3.jar	arxmlutil75.dll
commons-digester-1.7.jar	cmdbapi75.dll
commons-lang-2.2.jar	cmdbjni75.dll
log4j-1.2.14.jar	icudt32.dll
oncrpc.jar	icuinbmc32.dll
spring.jar	icuucbmc32.dll
	Xalan-Cbmc_1_9.dll
	XalanMessagesbmc_1_9.DLL
	xerces-cbmc_2_6.dll
	xerces-depdombmc_2_6.dll

Note:

- The AR System Java API is forward and backward compatible with other versions of the AR System. For a complete compatibility matrix, refer to the "API Compatibility" section in the *BMC Remedy/Atrium Developer Reference Guide*.
- The arencrypt*.dll files are only required if encryption is enabled on the Remedy server.

- b. Edit the **WrapperGateway.conf** file (or **WrapperManager.conf** if the Probe Manager and Gateway are running in separate mode) in the following directory:
C:\hp\UCMDB\DataFlowProbe\bin.

Add the following line after the **wrapper.java.library.path.2=%content_dll%** line:

```
wrapper.java.library.path.3=%runtime%/probeManager  
/discoveryResources/AtriumPushAdapter
```

- c. **For Atrium 7.6.04 and earlier versions only:** Add the complete path to the Atrium DLL files (for example, **C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\AtriumImportAdapter**) to the Windows System Path on the Data Flow Probe machine.
- d. Restart the Data Flow Probe service.

2. Prerequisites - Set up protocol credentials

Configure a generic protocol with the ARS server's username and password.

Note: While creating the generic protocol, set the protocol description to **atrium**.

3. Prerequisites - Create XML mapping files

This step involves creating XML mapping files (in the **<probe>\runtime\probeManager\discoveryResources\TQLEXP\Atrium\data** directory). These files map the BMC Atrium classes, attributes and relationships to their UCMDB equivalents. To create the XML mapping files for the topology requires identification of the topology to be imported from Atrium, and ensuring an equivalent topology exists in UCMDB. For more details, see "[Mapping Files](#)" on page 19.

4. Run the job - UCMDB 9.04 and later

In DFM, in the Integration Studio, create a new integration point.

- a. Provide a name and description for the integration point.
- b. Under **Integration Properties > Adapter**, select the **Population from Atrium** adapter.
- c. Configure the following adapter properties:
 - i. **ARS_Server**
 - ii. **ARS_Port**
 - iii. **BMC_NameSpace**
- d. Under **Adapter Properties > Probe Name** select the **Data Flow Probe** which will be used for the integration.
- e. Under **Adapter Properties > Trigger CI instance** select:
 - i. **Select Existing CI** (if you have a valid, existing CI). The **Select Existing CI** pane appears. Select the CI, or
 - ii. **Create New CI** (if you need to create a new CI). The **Topology CI Creation Wizard**

appears. Complete the creation of the CI using the Wizard.

Note: For details on the Topology CI Creation Wizard, see "Topology CI Creation Wizard" in the *HP Universal CMDB Data Flow Management Guide*.

- f. Save the integration point.
- g. Run the job.

Note: For details on running an integration job, see "Integration Studio" in the *HP Universal CMDB Data Flow Management Guide*.

5. Run the job - UCMDB 9.03 and 9.02

- a. Configure the following attributes for the job Import data from Atrium:
 - i. **ARS_Server**
 - ii. **ARS_Port**
 - iii. **BMC_NameSpace**
- b. Run the **Import data from Atrium** job.

Atrium Push Job

Adapter	15
Used Scripts	15
Parameters	16
Integration Flow	16

Adapter

This discovery uses the adapter called **Data Push into Atrium**.

Used Scripts

pushToAtrium.py

Parameters

Parameter	Description
credentialsId	The credentials ID to use for Atrium connection.
host	The host name or IP address of the remote Atrium server
port	The Atrium server's connection port (if not using portmapper)
probeName	An internal setting which UCMDB will automatically replace.

Integration Flow

Integration includes the following activities:

1. **Querying the UCMDB for CIs and relationships.** When an ad-hoc integration job is run in the Integration Studio, the integration process:
 - a. Receives the names of the integration queries that are defined in the job definition for that integration point.
 - b. Queries UCMDB for the results (new, updated, or deleted CIs and relationships) of these defined queries.
 - c. Applies the mapping transformation according to the pre-defined XML mapping files for every query.
 - d. Pushes the data to the Data Flow Probe.
2. **Sending the data to BMC Remedy/Atrium.** On the Data Flow Probe, the integration process:
 - a. Receives the CI and relationship data sent from the UCMDB Server.
 - b. Connects to the BMC Remedy/Atrium server using the Java API.
 - c. Transfers the CIs and relationships.

Import Data from Atrium Job

Adapter	17
Input CIT	17
Used Scripts	17
Discovered CITs	17
Parameters	18
Integration Flow	18

Adapter

This discovery uses the adapter called **Population from Atrium**.

Input CIT

The input CIT for this adapter is - **discoveryprobegateway**. The job uses an instance of the Discovery Probe Gateway which has access to connect to the remote BMC Atrium server.

Used Scripts

The adapter uses the following scripts:

Script	Description
atrium_query.py	Used to query BMC Atrium for data.
atrium_map.py	Used to map the queried data into data UCMDB can use.
atrium_to_ucmdb.py	Used to push imported data into UCMDB.

Discovered CITs

This integration can discover any CIT or relationship which is (a) mapped in the integration and (b) can be queried and converted to its UCMDB equivalent.

Parameters

Parameter	Detail
ARS_Port	The port for connecting to the ARS server. If portmapper is being used, this should be left as 0. Otherwise, specify the TCP port.
ARS_Server	The hostname or IP address of the BMC ARS server.
BMC_NameSpace	The BMC NameSpace to use. (For example: BMC.CORE.)
ChunkSize	The chunk size in which data should be retrieved from the remote server.
DateParsePattern	Set the date pattern to parse Atrium date strings.
DebugMode	Set to true to run integration in debug mode; this does not send data to UCMDB

Integration Flow

The Population from Atrium integration adapter flow has the following steps:

1. **Querying the Atrium server**

In this step, the integration adapter connects to the Atrium server and queries it for classes, attributes and relationships, described in the XML mapping files. The result of this step is the creation of intermediate XML files (in the `<probe>\runtime\probeManager\discoveryResources\TQLExport\Atrium\inter` directory).

2. **Mapping the data**

In this step, the data collected from the previous step and stored in the intermediate XML file, is converted into the UCMDB data format based on the mappings defined in the XML mapping files.

3. **Pushing the data to the UCMDB server**

In this final step, after being mapped into the UCMDB object state holder vector format, the data is sent to the UCMDB server.

Mapping Files

This section includes:

Mapping Files Overview	20
Mapping File Structure	20
Mapping File Elements	21
Main Parent Elements	21
CI Type Mapping Elements	21
Relationship Type Mapping Elements	23

Mapping Files Overview

A mapping file is an XML file that defines which CIT or relationship in UCMDB is mapped to which CIT or relationship in the target data store.

Mapping files:

- Control which CITs and relationships are to be pushed.
- Control the attributes for the CITs and relationships that are to be mapped.
- Map attribute values from multiple CIs to one target CI.
- Map attributes of children CIs (those having a **containment** or **composition** relationship) to the parent CI in the target data store. For example:
 - Set a **Number of CPUs** value for a target **node** CI.
 - Set a **Total Memory** value for a target **node** CI.
- Map attributes of parent CIs (those having a **containment** or **composition** relationship) in the target data store CI. For example, in the Atrium target data store, set the value of a **Container Server** attribute on the **Installed Software** CIT by retrieving the value of the UCMDB **Installed Software** CI container node.

Mapping File Structure

Every mapping file has the following skeletal structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<integration>
  <info>
    <source ... .. />
    <target ... .. />
  </info>
  <source_ci_type name="...">
    <target_ci_type name="...">
      <targetprimarykey>
        <pkey>...</pkey>
      </targetprimarykey>
      <target_attribute name="..." datatype="..." >
        <map type="..." />
      </target_attribute>
    </target_ci_type>
  </source_ci_type>
</integration>
```

Note: An elipsis (...) signifies a configurable section.

Mapping File Elements

This section includes:

- Main Parent Elements
- CI Type Mapping Elements
- Relationship Type Mapping Elements

Main Parent Elements

- **<integration>**. The root element of the XML file. This element has no attributes.
- **<info>**. The source and target data stores being used, for example:

```
<info>
<source name="Atrium" versions="7.6" vendor="BMC" />
<target name="UCMDB" versions="9.0" vendor="HP" />
</info>
```

- **<targetcis>**. The element that encapsulates the mapping for all CI types.
- **<targetrelations>**. The element that encapsulates the mapping for all relationship types.

CI Type Mapping Elements

- **<source_ci_type>**. The element that defines a CI type of the source data store, for example:

```
<source_ci_type name="BMC_ComputerSystem" nameSpace="BMC.CORE" query="">
```

- **Attribute: name.** Defines the name of the source CI type.
- **Attribute: mode.** Defines the mode of the update in the target data store.

- **<target_ci_type>**. The element that defines the target CIT, for example:

```
<target_ci_type name="unix">
```

- **Attribute: name.** Defines the name of the target CIT.

- **<targetprimarykey>**. The element that defines a list of all primary keys of the target CIT, for

example:

```
<targetprimarykey>
  <pkey>host_key</pkey>
</targetprimarykey>
```

- **<target_attribute>**. The element that defines an attribute mapping from the source CI type to the target CI type attribute. Attribute mapping can be of the following types:

- **Constant**. This type enables setting a constant value on the target attribute:

```
<target_attribute name="data_note" datatype="string" length="127">
  <map type="constant" value="ATRIUM DATA" />
</target_attribute>
```

- **Direct**. This type enables setting a direct value of a source data store attribute on the target data store:

```
<target_attribute name="name" datatype="string">
  <map type="direct" source_attribute="Name" />
</target_attribute>
```

- **Compound String**. This type enables the use of the above mapping types together to form more complex values for the target attribute, for example:

```
<target_attribute name="Bunch_0_Data" datatype="char" length="510" option="
uppercase">
  <map type="compoundstring">
    <source_attribute name="name"/>
    <constant value="_UNIX_Server, IP="/>
    <childattr name="ip_address" source_attribute="ip_address" aggregation=
"csv"/>
    <constant value=", CPU="/>
    <childattr name="cpu" source_attribute="display_label" aggregation="cs
v"/>
  </map>
</target_attribute>
```

Relationship Type Mapping Elements

- **<link>**. The element that defines a relationship mapping from the source data store to a target data store, for example:

```
<link source_link_type="composition"
      target_link_type="BMC_HostedSystemComponents"
      source_ci_type_end1="unix"
      source_ci_type_end2="cpu"
      role1="Source"
      role2="Destination"
      mode="update_else_insert">
  <target_ci_type_end1 name="BMC_ComputerSystem"
    superclass="BMC_System" />
  <target_ci_type_end2 name="BMC_Processor"
    superclass="BMC_SystemComponent" />
  ... Relationship attribute mapping elements similar to the CI type attribute
  mapping elements ...
</link>
```

- **Attribute: source_link_type.** Defines the name of the source link.
- **Attribute: target_link_type.** Define the name of the target link.
- **Attribute: source_ci_type_end1.** The **End1** CI type of the source link.
- **Attribute: source_ci_type_end2.** The **End2** CI type of the source link.
- **<target_ci_type_end1>.** Used to specific the value of the target links end1 CI type
- **<target_ci_type_end2>.** Used to specific the value of the target links end2 CI type

Troubleshooting and Limitations

The integration mapping file enables the mapping only of concrete CI types and relationships to the CI types and relationships in BMC Remedy/Atrium. That is, a parent CIT cannot be used to map children CIs. For example, if **UCMDB Node** is mapped to **BMC_ComputerSystem**, any Node CIT of type **Unix** is not transferred. A mapping must be separately created for **Unix** to **BMC_ComputerSystem**.