
Exporting the UCMDB Class Model to a UML Tool

This document includes the main concepts, tasks, and reference information for exporting the HP Universal CMDB class model to a Unified Modeling Language (UML) tool.

Note: Currently, Altova UModel versions 2008 and 2009 are supported.

This chapter includes:

Concepts

- Export to UML Tool – Overview on page 2

Tasks

- Export the Class Model on page 2
- Convert XML of Selected Part of Class Model on page 3

Reference

- Export Class Model to UML Wizard on page 4
- Tool Plug-in Input on page 7

Export to UML Tool – Overview

The Export to UML tool enables you to export selected sections of the UCMDB class model to a format compatible with UML tools, and to view the model as a UML diagram.

The input for the tool is the UCMDB class model XML file retrieved by the JMX service **Topaz/service=CMDB Class Model Services/exportClassModelToXml()** for UCMDB 8.x.

Export the Class Model

This task describes how to use the Export Class Model to UML tool wizard.

This task includes the following steps:

- “Prerequisites” on page 2
- “Run the UML Tool” on page 2
- “View the Exported File in Altova” on page 3

1 Prerequisites

Set up the tool by extracting **ExportToUML.zip** to any location. The file is located in the following directory:

<Volume>:\hp\UCMDB\UCMDBServer\j2f\tools

2 Run the UML Tool

You use the Export Class Model to UML wizard to choose the CIT and its attributes to be exported to the UML tool.

For details on working with the Export to UML wizard, see “Export Class Model to UML Wizard” on page 4.

3 View the Exported File in Altova

- a In Altova, select **File > Import From XMI File**.
- b Select the XMI file. The class model entry appears in the Model Tree pane.
- c Right-click the entry and select **Show in new diagram > Content**. Click **OK**.

Convert XML of Selected Part of Class Model

This task describes how to use the custom plug-in to convert the XML of a selected part of the class model to the UML tool format.

The plug-in can be either a Java class or an XSLT text file. If you use a Java class, it must implement the **ITransformToUML(ExportToUML.jar)** interface; the **transformToUML()** method receives an XML string as a parameter and returns a byte array that is written to the output file.

For details on the custom plug-in input, see “Tool Plug-in Input” on page 7.

This task includes the following steps:

- “Activate the Plug-in Using a Java Class” on page 3
- “Activate the Plug-in Using an XSLT Text File” on page 4

Activate the Plug-in Using a Java Class

Add the display name and fully-qualified class name to the **config.xml** file in the ExportToUML directory, as follows:

```
<ConverterToUML>
  <Name><display name></Name>
  <Class><fully qualified class name></Class>
</ConverterToUML>
```

Continue with the procedure for exporting the class model. For details, see “Export the Class Model” on page 2.

Activate the Plug-in Using an XSLT Text File

In this case, XSLT from the specified file is applied to the class model XML and is written to the output file.

Add the display name and full path of the XSLT file to the **config.xml** file in the ExportToUML directory, as follows:

```
<ConverterToUML>
  <Name><display name></Name>
  <XsltFile><Full_path_of_the_XSLT_file></XsltFile>
</ConverterToUML>
```

Continue with the procedure for exporting the class model. For details, see “Export the Class Model” on page 2.

Export Class Model to UML Wizard

Description	Enables you to choose the CIT and its attributes to be exported to the UML tool. To access: Browse to the location where you extracted the tool. Right-click ExportToUML.jar in the ExportToUML directory. Select Open With > Java 2 Platform SE binary .
Important Information	By default the tool opens the ClassModel.xml file in the ExportToUML directory. You open other files through the File > Open Class Model File menu.
Included in Tasks	“Export the Class Model” on page 2
Wizard Map	The Export Class Model to UML wizard contains: Export Class Model to UML Wizard > Select Valid Links > Export Selected Items
Useful Links	“Tool Plug-in Input” on page 7

The following elements are included:


GUI Element (A-Z)	Description
File	Displays a list of XML files. Open the file that contains the class model.
Select Attributes Pane	Select the attributes to be displayed in the UML tool. The attributes of the selected CIT are displayed in this pane. The attributes are colored as follows: <ul style="list-style-type: none"> ▶ Attributes inherited from the parent CIT are black. ▶ Attributes specific to this CIT are dark blue.
Select CI Types Pane	The CITs displayed here are retrieved from the ClassModel.xml file. Select the CIT and its sub nodes to be exported to the UML tool. Use the buttons to select or clear all subindex.



Select Valid Links

Description	Enables you to select links that can be used between two CITs. To access: Click the Next button in the Export Class Model to UML wizard.
Important Information	General information about the wizard is available in “Export Class Model to UML Wizard” on page 4.
Included in Tasks	“Export the Class Model” on page 2
Wizard Map	The Export Class Model to UML wizard contains: Export Class Model to UML Wizard > Select Valid Links > Export Selected Items

The following elements are included (unlabeled GUI elements are shown in angle brackets>):

GUI Element (A-Z)	Description
Filter by	Enables you to filter the list of valid links. You can filter the list of CITs and you can sort the list by CIT or relationship.
	Click to remove the filter from the list of CITs.

Export Selected Items

Description	Enables you to export the selected CITs and their attributes to the UML tool. To access: Click the Next button in the Select Valid Links page.
Important Information	General information about the wizard is available in “Export Class Model to UML Wizard” on page 4.
Included in Tasks	“Export the Class Model” on page 2
Wizard Map	The Export Class Model to UML wizard contains: Export Class Model to UML Wizard > Select Valid Links > Export Selected Items

The following elements are included (unlabeled GUI elements are shown in angle brackets>):

GUI Element (A-Z)	Description
Export	Click to browse to the UML file, in XMI format, to which you want to add the UCMDDB CITs and their attributes. If there are no valid links for a pair of CITs, a no valid links message is displayed in the Select Valid Links page.

Tool Plug-in Input

The input for the plug-in is an XML string (selected classes/attributes/valid links) in the following form:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Class-Model>
  <Class class-name="hostresource" display-name="Host Resource"
visibility="public">
    <Attribute name="isvirtual" display-name="Is Virtual" visibility="public"/>
    <Attribute name="city" display-name="City" visibility="public"/>
  </Class>
  <Class class-name="host_node" display-name="Computer" visibility="public">
    <Derived-From class-name="host"/>
  </Class>
  <Class class-name="vax" display-name="VAX" visibility="public">
    <Attribute name="root_actualdeletionperiod" display-name="Actual Deletion
Period" visibility="public"/>
    <Attribute name="data_allow_auto_discovery" display-name="Allow CI Update"
visibility="public"/>
    <Derived-From class-name="host_node"/>
  </Class>
  <Class class-name="host" display-name="Host" visibility="public">
    <Attribute name="host_iscomplete" display-name="Host Is Complete"
visibility="public"/>
    <Attribute name="host_isroute" display-name="Host Is Route" visibility="public"/>
    <Attribute name="host_hostname" display-name="Host Name" visibility="public"/>
    <Attribute name="host_os" display-name="Host Operating System"
visibility="public"/>
  </Class>
  <Class class-name="unix" display-name="Unix" visibility="public">
    <Derived-From class-name="host_node"/>
  </Class>
  <Valid-Link ID="host_member_host" display-name="Member" visibility="public">
    <End1 class-name="host"/>
    <End2 class-name="host"/>
  </Valid-Link>
  <Valid-Link ID="host_container_f_hostresource" display-name="Container link"
visibility="public">
    <End1 class-name="host"/>
    <End2 class-name="hostresource"/>
  </Valid-Link>
</Class-Model>
```

