



CHAPTER 5

Using Command Modeler

Command Modeler provides an infrastructure for generating and validating device-independent CLI models. Developers can use the generated device independent CLI models to generate device-specific Java code.

The generated Java code can be used in Domain Manager applications as a replacement to the hard coded template-based CLI generation mechanism that is prevalent.

Command Modeler is a tool that is built over the EDI Device CLI Knowledge Base. The code generated by Command Modeler also takes care of device-specific capability and range variations. For example, the number of VLANs that can be created, the maximum number of MAC addresses that can be learned on a switch port when security is enabled, etc.)

The device-independent virtual model is also called the Unified CLI View or Device Independent CLI Model.

The Unified CLI View is a unionized representation of all the CLIs that are available in a set of devices/IOS versions that the you select. Command Modeler support is only available for IOS in this release.

Command Modeler works on both Linux and Windows platforms.

To use Command Modeler, see these topics:

- [Getting Started With Command Modeler](#)
- [Understanding the Command Modeler Workflow](#)
- [Understanding the Command Modeler UI](#)
- [Creating a Model and Generating Code](#)
- [Sample Model File](#)

Getting Started With Command Modeler

Command Modeler is an Eclipse-based UI tool. It is packaged with Cisco E-DI, and will be available after you have installed Cisco E-DI. To install Cisco E-DI, see:

- *Installation Guide for Enhanced Device Interface, 2.2 on Windows*
- *Installation Guide for Enhanced Device Interface, 2.2 on Linux*

To get started with Command Modeler, see these topics:

- [Pre-requisites](#)
- [Launching Command Modeler](#)

- [Configuring Command Modeler](#)

Pre-requisites

To work with Command Modeler you should have device packages available. If you have installed Cisco E-DI, the device packages packaged with this release of Cisco E-DI are available in this device package directory:

E-DI Install Location\Cisco Systems\Cisco EDI\edi\dist\devpackages

For the device packages provided with this build, see the E-DI 2.2 Release Notes.

Additional device packages or Incremental Device Updates (IDUs) will be made available through Cisco.com. IDUs allow Cisco E-DI to be updated with support for new device packages. To download the files for the device packages, see <http://www.cisco.com/public/sw-center/sw-netmgmt.shtml>.

After the required IDUs are downloaded, they can be copied to Cisco E-DI's devpackage directory.

See Cisco E-DI Installation and Setup Guide 2.2, for details of the installation process.

Launching Command Modeler

Before you launch Command Modeler, from the command prompt, navigate to your Eclipse folder (where your eclipse.exe) resides. Run the command `eclipse -clean`. This ensures that the cache is cleaned.

Follow these steps to launch Command Modeler:

-
- Step 1** Navigate to the folder *E-DI Install Location*\Cisco EDI\edi\dist\cmdModeler\lib.
 - Step 2** Copy the Command Modeler jar file (`com.cisco.edi.cmdModeler_buildnumber.jar`) from the lib folder.
 - Step 3** Paste the jar file into the Eclipse plugins folder (*Eclipse Directory*\plugins).



Note Ensure that Eclipse is closed while the jar is copied into the plugins folder.

- Step 4** Start Eclipse.
- Step 5** From the Main Menu Bar, select **Windows > Open Perspective > Other**.

The Open Perspective dialog box opens.

Select **Command Modeler** and click **OK**.

The Command Modeler perspective opens.

For a description of the Command Modeler views, see [Understanding the Command Modeler UI](#).

Your next task is to configure Command Modeler. See [Configuring Command Modeler](#)

Configuring Command Modeler

To be able to begin working with Command Modeler, you should configure Command Modeler for certain default settings such as the device packages directory and the output directory (where the generated code and the XML Model file will be stored).

To configure Command Modeler:

-
- Step 1** From the Main Menu Bar, select **Command Modeler > Setup Wizard**.
The Command Modeler Setup Wizard opens.
- Step 2** In the Device Packages Dir. field, enter the location of your device packages.
Or
Use **Browse** to select the directory in which your device packages reside. The default directory is *E-DI Install Location\Cisco Systems\Cisco EDI\edi\dist\devpackages*.
- Step 3** In the Code Output Dir. field, enter the location where you want to save your device model output (generated code or XML file).
You can use **Browse** to select the directory in which your device model output (the Java code that you create) should reside. Alternatively, click **Make New Folder**, and create a new folder for your output.
- Step 4** Click **Finish**.
-

Understanding the Command Modeler Workflow

The following is the high-level workflow of Command Modeler:

-
- Step 1** Configure Command Modeler by specifying the device package directory and the output directory. See [Configuring Command Modeler](#)
- Step 2** Specify a name for the model and select the devices and versions for which you need to create a Unified CLI View.
A Command Modeler Unified CLI View is created and all the sub-modes/views of the Unified CLI View are displayed in a tree format in the Commands Tree View. See [Creating a Model and Generating Code](#).
You can view the details of each node using the Node Details View.
- Step 3** Insert the commands into the Pre-Model by selecting a command and using the Add Menu option for the particular node, in the Tree View. See [Creating a Model and Generating Code](#).
- Step 4** Generate the Model (XML file) from the Pre-Model that you created.
This is a read-only file. However, you can select **Save As** and save it with an XML extension. You can save this model under an existing Java project, and open it at any time to make any changes to the model. See [Creating a Model and Generating Code](#).
- Step 5** Generate device-specific Java code using the Generate Code option from the Pre-Model view.
Command Modeler creates and saves the Java files to the output directory that you specified when you configured Command Modeler.

To check for any information or errors during the Java code generation, see [Troubleshooting Command Modeler](#).

Understanding the Command Modeler UI

To launch the Command Modeler Perspective from the Eclipse Menu Bar, see [Launching Command Modeler](#).

The following are the component views of the Command Modeler Perspective:

- [Commands Tree View](#)
- [Node Details View](#)
- [Pre-Model List View](#)

Commands Tree View

The Tree View displays all the submode views from the Unified CLI view. You can drill down to see the commands under each submode and nodes within each command.

One of the main submodes is the Configure submode.

The Interface command is not available under Configure submode.

interface *Interface Name* command will be automatically added when you add any command from the appropriate interface submode view.

Node Details View

When you select a node from the Tree View, the Node Details view shows the validity of the node under the device-version pair.

The Node details are displayed for all the device-version pairs in the Unified CLI View that was created.

The following details are displayed for each node:

Field	Description	Example (for command <code>aggregate-address</code>) for Cat 3550 and Cat3750
Name	Name of the node.	A.B.C.D
Type	The E-DI representation for the node. For example, key word node, number node, IP address node, etc.	IpAddressNode
Desc	Description of the selected node.	Aggregate address
Min	Minimum value applicable.	--NA--
Max	Maximum value applicable.	--NA--
Sample	Sample value.	0.0.0.0

Pre-Model List View

The Pre-Model list is created when you right-click a submode or node from the Tree View and select **Add** from the context menu that appears. The selected node is added to the Pre-Model List.

In the Pre-Model List view, the command, with the sample values, and the View Name (the submode under which the node was selected) are displayed.

You will not be allowed to insert a command into the Pre-Model List if it does not contain a `<cr>` (line termination).

If you select a command from any view other than the `config` view, Command Modeler automatically inserts entry commands to that view from the `config` mode.

For example if you add a command such as `network A.B.C.D` under `config-rip`, Command Modeler will automatically enter the command `router rip` of Configure submode to the Pre-Model List. Similarly Command Modeler adds `exit` commands where relevant.

If a command that you add is not present in the current sub-mode, then Command Modeler adds Exit commands and Entry commands as required.

To remove a command, right-click and select **Remove** or **Delete**. The command will be removed from the Pre-Model List.

For the procedure to create a Model from the Pre-Model List and to generate the Command Modeler code, see [Creating a Model and Generating Code](#).

Creating a Model and Generating Code

Before you begin the task of creating a Model, ensure that you have configured Command Modeler. See [Configuring Command Modeler](#).

You can create a device-independent virtual model. To do this:

-
- Step 1** From the Main Menu Bar, select **Command Modeler > Unified-View Creation Wizard**.
The Unified-View Creation Wizard opens.
The Devices List displays all the available device packages in the directory that you had specified.
 - Step 2** From the Devices List, select the device-version pairs. You should select at least two device-version pairs to be able to create the Unified-View.
 - Step 3** Enter the Model name in the Model Name field. This can be any alphanumeric string of your choice.
 - Step 4** Click **Finish**.
After the selected device packages are loaded into the memory, the Commands Tree View on the left, is refreshed with the unified submodes. This may take a few seconds.
For details about the Commands Tree View, see [Understanding the Command Modeler UI](#).
 - Step 5** Right-click on node from the Commands Tree View and select **Add** from the context-menu.
The selected node is added to the Pre-Model List. For details about the Pre-Model List, see [Understanding the Command Modeler UI](#).
You can add more than one command to the Pre-Model List.
 - Step 6** Right-click in the Pre-Model List area and select **Create Model**.

The Model, an XML representation of the selected commands is displayed in the Editor view. This XML file is not editable.

However if you want to save this file and edit it later, select **File > Save As** and save the XML file with a .xml extension. You can review this file, or treat it as a draft, make changes to it or to the Pre-Model List until you have all the required commands, and then create the model again.

After all your changes are done you can use this XML file to generate code.

Step 7 Right-click in the Pre-Model List view and select **Generate Code**.

If you had previously saved an XML file and have reopened it in the Editor to make changes, then you can right-click in the Editor and select **Command Modeler > Recreate Model**. It will rebuild the device package environment in which the model file was saved.

Add additional commands as needed. After completing the changes, you can right-click in the Editor and select **Command Modeler > Generate Code**.

After the code is generated, the following message is displayed to show the success of the code generation:

```
File was saved in the Output Directory: directory path
```

Where *directory path* is the path to the output directory that you specified at the time of configuring E-DI.

Command Modeler creates Java interface and implementation files in the output directory.

It creates one Java file for each of the device versions that you selected, and one interface file. The implementation files follow the naming convention of `devicenam_OSversion_modelname`

For example, if you had selected Cat3550-12.1(14)EA1a, Cat3550-12.1(22)EA2 and Cat3750-12.1(19)EA1a, you will see the following Java files in your output folder:

- `Cat3750_12_1_19_EA1a_ModelName.java`
- `Cat3550_12_1_22_EA2_ModelName.java`
- `Cat3550_12_1_14_EA1a_ModelName.java`

and

- `I_ModelName.java`

where *ModelName* is the name of the model that you specified, and the prefix *I* indicates an interface file.

In the device-specific implementation files, set and check methods are implemented. `toCli` method returns a string which represents the commands for the model. `negCli` method returns a string which represents negation commands for the model.

The check method checks for the validity for a specific node.

Troubleshooting Command Modeler

You can view the information, errors, or warnings that may arise during XML file generation using the Eclipse Error Log.

To invoke the Eclipse Error Log:

Step 1 From the Main Menu Bar, select **Window > Show View > Other**.

The Show View dialog box appears.

Step 2 Select **PDE Runtime > Error Log**.

The Error Log appears.

The Error Log contains information, error and warning messages from all components. It is displayed under the columns Message, Plug-in, and Date:

- The Message column displays the message.
- The Plug-in column indicates the specific component that has generated the messages
- The Date column displays the date when the message was generated.

For troubleshooting purposes, you can export the log using the Export Log option (icon) of Eclipse, and send it to customer support.

Launching the Command Modeler Help

To launch the help:

Step 1 Navigate to *E-DI Install Location*\Cisco EDI\edi\dist\cmdModeler\lib.

Step 2 Copy the jar file com.cisco.edi.cmdModeler.doc.help.jar.

Step 3 Paste the jar file into the Eclipse plugins folder (*Eclipse Directory*\plugins).



Note Ensure that Eclipse is closed while the jar is copied into the plugins folder.

Step 4 Start Eclipse.

Step 5 From the Eclipse Menu Bar, select **Help > Help Contents**.

The help opens.

Sample Model File

The following is a sample Model file of Command Modeler:

In this sample, the format enclosed within the tags `<command>` and `</command>` has the Command ID and the View Name. If the View is not valid in a particular device it will not be added into the persisted file.

```
<model>

  <command>
    <id><![CDATA[router rip]]></id>
    <viewName>Cat2950-configure</viewName>
    <viewName>Cat3550-configure</viewName>
  </command>

  <command>
    <id><![CDATA[network 0.0.0.0]]></id>
```

```

        <viewName>Cat3550-configure-rip</viewName>
    </command>

    <command>
        <id><![CDATA[exit]]></id>
        <viewName>Cat3550-configure-rip</viewName>
    </command>
</model>

```

Sample XML Output File

The following is a sample XML output file of Command Modeler:

```

<model>

    <command>
        <id><![CDATA[router bgp 10]]></id>
        <viewName>Cat2950-configure#12.1(13)EA1c</viewName>
        <viewName>Cat3550-configure#12.1(14)EA1a</viewName>
    </command>

    <command>
        <id><![CDATA[network 0.0.0.0]]></id>
        <viewName>Cat3550-configure-bgp#12.1(14)EA1a</viewName>
    </command>

    <command>
        <id><![CDATA[exit]]></id>
        <viewName>Cat3550-configure-bgp#12.1(14)EA1a</viewName>
    </command>
</model>

```