



CHAPTER 11

Creating and Using Macro Commands

The Macro command feature of E-DI helps you to define device and OS-independent commands.

A Macro command provides an abstraction for command sets and/or command variations. In a single operation, you can download commands to various types of platforms that have variation in CLI commands. In this way, Macro commands make it easier for you to deal with variations across OS versions and platforms for configuration changes.

This feature is available for IOS, Cat OS and PIX platforms.

E-DI contains simple macros for a set of commands that have different variations on different platforms. These Macro commands are packaged along with E-DI. You can also create your own macros.

The macros contain simple keywords and command parameters similar to template variables. If you enter these keywords and parameters, the internal syntactical and semantic differences among various devices, OS types, and versions are handled internally by E-DI.

Each Macro command that is defined can be translated to commands on all available device families and OS types. They can also be translated to a particular device family and OS type.

When a corresponding configlet for a particular device family is defined, E-DI verifies and translates this. If there are errors during translation, the particular macro support for that device family will not be available.

E-DI compiles the macro into macro XML files, based on the device family. E-DI then converts the macro XML files to CLI. This translation makes the command variations transparent to you.

You can create and use macros:

- Through CLI (see [Creating and Using Macros Through CLI](#))
- Through GUI (see [Creating and Using Macros Through UI](#))

Creating and Using Macros Through UI

Macro Command Manager is an Eclipse- based, standalone UI tool of E-DI. It allows you to create, edit, delete, and deploy macros to devices.

To launch macro Command Manager, see [Launching Macro Command Manager](#)

Launching Macro Command Manager

Macro Command Manager 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 and Setup Guide for Enhanced Device Interface, 2.2 on Windows*
- *Installation Guide and Setup Guide for Enhanced Device Interface, 2.2 on Linux*

**Note**

Before you start Macro Command Manager, ensure that the EDI Service is running. To start this service, On Windows, go to **Start > Programs > Cisco E-DI > E-DI Service > Start**. On Linux, navigate to *E-DI Install Location* /Cisco-EDI/bin and enter `./start` at the command prompt.

Before you launch Macro Command Manager: (These steps are optional.)

Step 1 Go to the command prompt and navigate to your Eclipse folder (where `eclipse.exe` resides).

Step 2 Run the command `eclipse -clean`
This ensures that the cache is cleaned.

Follow these steps to launch Macro Command Manager:

Step 1 On Windows:

- Choose **Start > Programs > Cisco E-DI > E-DI Service > Device Configuration Manager**.
- or
- Navigate to the directory *E-DI Install Location*\Cisco EDI\edi\dist\ui_products\configmanager and double-click on launcher .exe.

On Linux:

- Navigate to *E-DI Install Location*/Cisco EDI/edi/dist/ui_products/configmanager and enter `./launcher`

Step 2 E-DI prompts you to log in to the E-DI server.

By default, DCM connects to the port 2323 on the server.

If the server Telnet port is not 2323, you should change this value in the **eclipse.ini** file. This file is located in the following location:

On Windows:

E-DI Install Location\edi\dist\ui_products\configmanager.

On Linux:

E-DI Install Location/edi/dist/ui_products/configmanager

Step 3 After you log in, Device Configuration Manager opens. This has three perspectives (which appear as buttons):

- Config Manager (default perspective, which is highlighted.)
- Macro Command Manager
- Command Translator

Step 4 Click **Macro Command Manager**.

The Macro Command Manager perspective opens and you see the Device Drawer in the left pane.

Alternatively, to launch Macro Command Manager from the Config Manager perspective, you can click the Other icon and select Other.

The Open Perspective dialog box appears.

Select Macro Command Manager and click **OK**.

The Macro Command Manager perspective opens.

Understanding the High-level Workflow of Macro Command Manager

This is the high-level workflow of the Macro Command Manager:

Step 1 Create a Macro Package.

A Macro Package is a container for a set of macros. It allows you to easily organize your Macros. See [Creating a Macro Command](#).

Step 2 Create a Macro.

A macro contains configlets, parameters and their descriptions. [Creating a Macro Command](#).

Step 3 Create Configlets.

A Configlet is a command holder for device variation. It is a set of related commands meant to do a specific job on devices. [Creating a Macro Command](#).

Step 4 Deploy the macro.

See [Deploying a Macro Command to a Device](#).

Understanding the Macro Command Manager UI

To launch Macro Command Manager, see [Launching Macro Command Manager](#).

Macro Command Manager has the following views:

- Macro Command Tree View

Displays the Macro packages, macros and configlets in a hierarchical view. A default Macro package, Root, always exists. If you right-click in the tree view, a context menu appears with these options:

- Deploy Macro
- Add Macro Package
- Delete Macro Package
- Add Macro
- Delete Macro
- Add Configlet
- Delete Configlet

- Macro View

Displays the Macro details such as the Package to which it belongs, the Macro name, the parameters and their description.

- Macro Command Editor and Config Editor View

These views appear only when you select a Macro from the Tree view, right-click on it and then select Add Configlet.

You can enter the details of a configlet here. The Config Editor allows you to enter commands. You can also select commands using the Config Editor.

Creating a Macro Command

To create a Macro Command you must:

1. Create a Macro Package—see [Creating a Macro Package](#).
2. Create a Macro—see [Creating a Macro](#).
3. Create a Configlet—see [Creating a Configlet](#).

Creating a Macro Package

A default Macro Package, Root, exists in the Macro Command Tree View. You can add macros to this or create a new package.

To create a new Macro Package:

-
- | | |
|---------------|--|
| Step 1 | Go to the Macro Command Tree View, right-click and select Add Macro Package from the context-menu that appears. |
| Step 2 | Enter the package name. |
| Step 3 | Enter a short description for the package. |
| Step 4 | Click OK . |
| Step 5 | The new Macro Package appears in the Macro Command Tree View. |
-

To delete a Macro Package, select the package, right-click on it and select **Delete Macro Package** from the context menu that appears.

Creating a Macro

To add a Macro:

-
- | | |
|---------------|---|
| Step 1 | Go to the Macro Command Tree View, select a macro package, right-click and select Add Macro from the context-menu that appears.

The Macro Addition Wizard appears. |
| Step 2 | Expand Macro Basic Parameters.

The Package Name field is populated by default, with the selected parent package name.

The Root Package is represented by a forward slash (/). All other user created Macro Packages appear with this format: <i>\Package_name</i> . |

- Step 3** Enter the Macro name.
- Step 4** Expand Parameters List.
- Step 5** Enter a name for the Parameter, and add a description in the Description field.
- Step 6** Click **Add**.

The Parameter appears as a variable in the table below. The name and description details are displayed. Use the Add/Remove buttons to add another property or remove existing ones.

Click **Finish**.

To delete a Macro, select the macro, right-click on it and select **Delete Macro** from the context menu that appears.

Creating a Configlet

To create a Configlet:

- Step 1** Go to the Macro Command Tree View, select a macro, right-click, and select **Add Configlet**.
The Macro Config Editor and Command Editor views appear.
The Macro Configlet Index is automatically populated. This cannot be changed.
- Step 2** Select the Device Family from the drop-down list.
- Step 3** Select the IOS image version from the Version drop-down list.
- Step 4** Go to the Config Editor and enter the parameterized CLI commands that you want to include in the configlet.

When you enter the parameters, specify the type of the parameter by appending **:** and the hint after parameter name. This will be automatically done if you put a colon and enter Ctrl+Space.
For example, if you have defined **\$p1** as a parameter in the macro, you can use this parameter in the configlet as below:

`logging $p1:A.B.C.D`
- Step 5** Save the configlet by selecting it and then selecting **Configlet > Save Configlet** from the Main Menu.
The configlet appears in the Macro Command Tree View.

To edit an existing configlet, select it from the Tree View. The configlet details appear in the MacroCommand.Editor and Command Editor views. Change the details and save the configlet.

To delete a configlet, select it from the Tree View, right-click on it and select **Delete Configlet** from the context menu that appears.

Deploying a Macro Command to a Device

To deploy a Macro Command to a device:

-
- Step 1** Select a Macro Package, right-click on it and select **Deploy Macros** from the context-menu that appears. The Deploy Macros wizard appears.
- Step 2** Enter the values for the parameters in the Configlet.
- Step 3** Click **Associate Device** to associate devices to which the macros should be deployed. A pop-up appears.
- Step 4** Enter the Device IP and click **Go**.
Or
Select the devices from the Device Drawer.
- Step 5** Click **OK**.
- Step 6** Click **Next** in the Deploy Macros dialog box. The Job Scheduler dialog box appears. You can:
- Schedule a job to deploy the macros.
Enter a name for the job, select a date using the calendar icon, and select the time from the drop-down list box. The job will run and deploy the macros as scheduled.
or
 - Deploy the macros to the selected devices immediately. The macros will be deployed immediately.
- Step 7** Click **Finish**.
-

Creating and Using Macros Through CLI

Using NetCLI when you specify a macro, use this convention:

/packagename.macroname

where *packagename* is the name of the macro package and *macroname* is the name of the macro.

Table 11-1 **Commands to Create Macros**

Action	Command
To enter macro configuration mode. Displays a message: Entering Macro Configuration mode. In the macro configuration mode you have the following options: • compile— • delete • end • exit • export • import • macro • new • show	<code>user@server [SVR:/] # configure macro</code> Displays a message: Entering Macro Configuration mode. The prompt changes to <code>user@server [SVR:/<macro># ?</code> [compile delete end exit export import macro new show]
To create a macro package or a macro.	<code>user@server [SVR:/<macro>] # new [macro package]</code>
To create a new package.	<code>user@server [SVR:/<macro>] # new package <i>packagename</i></code> where <i>packagename</i> is a name of your new macro package This message appears: You are creating a new package. Entering Macro Configuration mode.
To create a new macro.	<code>user@server [SVR:/<macro>] # new macro <i>macroname</i></code> where <i>macroname</i> is a name of your new macro. You are creating a new macro. Entering Macro Configuration mode. The prompt changes to: <code>user@server [SVR:/] (macro-edit) # [configlet end exit keyword param show]</code>

Table 11-1 **Commands to Create Macros (continued)**

Action	Command
To create a new configlet.	<pre>user@server [SVR:/] (macro-edit) #configlet ConfigletIndex scope DeviceFamily ImageVersion</pre> <i>Configlet Index</i> is the index of the configlet that you are creating. Select from 1 to 99. <i>scope</i> is the scope of the configlet with regard to the device family and OS version. <i>DeviceFamily</i> is the device family. <i>ImageVersion</i> is the image version for that device family. This message appears: You are creating a new configlet. Entering Configlet Configuration mode. Now specify the commands for the configlet. Use ? to view and select commands. After you are done press Enter. The prompt changes to include the device family-OS version pair that you selected. For example, <pre>user@server [SVR:/] (Cisco7600-12.2(18)SXD4-configlet)</pre> Now specify the parameters, for example \$parameter 1, \$parameter 2, etc., and press Enter. EDI matches your configlet with the device packages. If successfully matched, you will see the result as *MATCHED*
To edit a macro.	<pre>user@server [SVR:/<macro>] #macro macroID</pre> where <i>macroID</i> is the name/ID of the Macro that you want to edit. These details of the Macro appear: • Keyword and description. • Parameters, and their description. Then the following message appears: Entering Macro <i>macroID</i> where <i>macroID</i> is the name/ID of the Macro that you want to edit The prompt changes to <code>user@server [SVR:/<macro-edit>]# [configlet end exit keyword param show]</code> To edit, follow the same processes used while creating a macro.
To convert the macro into the ED-I CLI XML models. After compilation, your XML file appears in this location: <i>CiscoEDI Install Location</i>/Cisco E-DI/edi/resources/server/macro where <i>CiscoEDI Install Location</i> is the location where E-DI is installed. This feature is not supported through GUI.	<pre>user@server [SVR:/<macro>] #compile packagename</pre> where <i>packagename</i> is a macro package. Upon successful compilation, the following message appears: Package Compiled into <i>CiscoEDI Install Location</i>/Cisco E-DI/edi/resources/server/macro

Table 11-1 **Commands to Create Macros (continued)**

Action	Command
To export a macro to an XML file. This feature is not supported through GUI.	<pre>user@server [SVR: /<macro>] # export macro_package_name XMLfilename</pre> where <i>macro_package_name</i> is the name of the macro package to be exported and <i>XMLfilename</i> is the name of the XML file to which the macro package needs to be exported. For example, if you enter: <pre>user@server [SVR: /<macro>] # export logging logging.xml</pre> the Macro package named <i>logging</i> is exported to the file <i>logging.xml</i>
To import a macro from an XML file. This feature is not supported through GUI.	<pre>user@server [SVR: /<macro>] # import filename</pre> where <i>filename</i> is the name of the XML file from which the macro needs to be imported.
To delete a package or a macro.	<pre>user@server [SVR: /<macro>] # delete macro packagename</pre> where <i>packagename</i> is the name of the package. or <pre>user@server [SVR: /<macro>] # delete macro macroname</pre> where <i>macroname</i> is the name of the macro.
To view macro packages, macro lists, or macro details.	<pre>user@server [SVR: /<macro>] # show [details list packages]</pre>
To view details of macro. For a given macro, these details are displayed: • Keyword and description. • Parameters, and their description.	<pre>user@server [SVR: /<macro>] # show details Macro ID</pre> where <i>Macro ID</i> is the name of the Macro. For example: <pre>user@server [SVR: /<macro>] # show details \log</pre> shows you the details about the macro with name/ID as <i>log</i>.
To view the list of Macros. These details appear in a tabular format: • Packages • The associated Macro IDs (names of macro) • Parameters associated with each Macro ID.	<pre>user@server [SVR: /<macro>] # show list</pre>
To view the list of macro packages. These details are displayed: • The list of macro packages • The number of macros associated with each package. The root package (default) is represented by /	<pre>user@server [SVR: /<macro>] # show packages</pre>

