



## Using the APIC CLI

---

Beginning with APIC release 6.1(2), the configuration mode of the Cisco APIC NX-OS-style CLI is no longer supported. Other functionality on the APIC CLI such as the EXEC mode and UNIX CLI are still available.

- [Accessing the NX-OS Style CLI, on page 1](#)
- [Using the NX-OS Style CLI for APIC, on page 2](#)
- [Differences in Usage from NX-OS, on page 5](#)
- [Limitations with mixing the user interfaces, on page 5](#)

## Accessing the NX-OS Style CLI



---

**Note** From Cisco APIC Release 1.0 until Release 1.2, the default CLI was a Bash shell with commands to directly operate on managed objects (MOs) and properties of the Management Information Model. Beginning with Cisco APIC Release 1.2, the default CLI is a NX-OS style CLI. The object model CLI is available by typing the **bash** command at the initial CLI prompt.

---

### Procedure

---

- Step 1** From a secure shell (SSH) client, open an SSH connection to APIC at *username@ip-address*.  
Use the administrator login name and the out-of-band management IP address that you configured during the initial setup. For example, `admin@192.168.10.1`.
- Step 2** When prompted, enter the administrator password.
- 

### What to do next

When you enter the NX-OS style CLI, the initial command level is the EXEC level. From this level, you can reach these configuration modes:

- To continue in the NX-OS style CLI, you can stay in EXEC mode or you can type **configure** to enter global configuration mode.

For information about NX-OS style CLI commands, see the *Cisco APIC NX-OS Style CLI Command Reference*.

- To reach the object model CLI, type **bash**.

For information about object mode CLI commands, see the *Cisco APIC Command-Line Interface User Guide, APIC Releases 1.0 and 1.1*.

## Using the NX-OS Style CLI for APIC

### Using CLI Command Modes

The NX-OS style CLI is organized in a hierarchy of command modes with EXEC mode as the root, containing a tree of configuration submodes beginning with global configuration mode. The commands available to you depend on the mode you are in. To obtain a list of available commands in any mode, type a question mark (?) at the system prompt.

This table lists and describes the two most commonly used modes (EXEC and global configuration) along with an example submode (DNS). The table shows how to enter and exit the modes, and the resulting system prompts. The system prompt helps to identify which mode you are in and the commands that are available to you in that mode.

Mode	Access Method	Prompt	Exit Method
EXEC	From the APIC prompt, enter <b>execsh</b> .	apic#	To exit to the login prompt, use the <b>exit</b> command.
Global configuration	From EXEC mode, enter the <b>configure</b> command.	apic(config)#	To exit from a configuration submode to its parent mode, use the <b>exit</b> command.
DNS configuration	From global configuration mode, enter the <b>dns</b> command.	apic(config-dns)#	To exit from any configuration mode or submode to EXEC mode, use the <b>end</b> command.

### CLI Command Hierarchy

Configuration mode has several submodes, with commands that perform similar functions grouped under the same level. For example, all commands that display information about the system, configuration, or hardware are grouped under the **show** command, and all commands that allow you to configure the switch are grouped under the **configure** command.

To execute a command that is not available in EXEC mode, you navigate to its submode starting at the top level of the hierarchy. For example, to configure DNS settings, use the **configure** command to enter the global configuration mode, then enter the **dns** command. When you are in the DNS configuration submode, you can query the available commands. as in this example:

```
apic1# configure
apic1(config)# dns
apic1(config-dns)# ?
  address  Configure the ip address for dns servers
  domain   Configure the domains for dns servers
  exit     Exit from current mode
```

```

fabric      Show fabric related information
no          Negate a command or set its defaults
show        Show running system information
use-vrf     Configure the management vrf for dns servers
where       Show the current mode

```

```

apic1(config-dns) # end
apic1#

```

Each submode places you further down in the prompt hierarchy. To view the hierarchy for the current mode, use the **configure** command, as shown in this example:

```

apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# where
configure t; bgp-fabric
apic1(config-bgp-fabric)#

```

To leave the current level and return to the previous level, type **exit**. To return directly to the EXEC level, type **end**.

### EXEC Mode Commands

When you start a CLI session, you begin in EXEC mode. From EXEC mode, you can enter configuration mode. Most EXEC commands are one-time commands, such as show commands, which display the current configuration status.

### Configuration Mode Commands

Configuration mode allows you to make changes to the existing configuration. When you save the configuration, these commands are saved across switch reboots. Once you are in configuration mode, you can enter a variety of protocol-specific modes. Configuration mode is the starting point for all configuration commands.

### Listing Commands and Syntax

In any command mode, you can obtain a list of available commands by entering a question mark (?).

```

apic1(config-dns) # ?
address      Configure the ip address for dns servers
domain       Configure the domains for dns servers
exit         Exit from current mode
fabric       Show fabric related information
no           Negate a command or set its defaults
show         Show running system information
use-vrf      Configure the management vrf for dns servers
where        Show the current mode

apic1(config-dns) # end
apic1#

```

To see a list of commands that begin with a particular character sequence, type those characters followed by a question mark (?). Do not include a space before the question mark.

```

apic1(config) # sh ?
aaa           Show AAA information
access-list   Show Access-list Information
accounting    Show accounting information

```

```

    aclog           Show aclog information
    . . .

```

To complete a command after you begin typing, type a tab.

```

apicl# qu<TAB>
apicl# quota

```

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help because it reminds you which keywords or arguments are applicable based on the commands, keywords, and arguments you have already entered.

```

apicl(config-dns)# use-vrf ?
inband-mgmt  Configure dns on inband
oob-mgmt     Configure dns on out-of-band

apicl(config-dns)#

```

You can also abbreviate a command if the abbreviation is unambiguous. In this example, the **configure** command is abbreviated.

```

apicl# conf
apicl(config)#

```

### Undoing or Reverting to Default Values or Conditions Using the 'no' Prefix

For many configuration commands, you can precede the command with the **no** keyword to remove a setting or to restore a setting to the default value. This example shows how to remove a previously-configured DNS address from the configuration.

```

apicl(config-dns)# address 192.0.20.123 preferred
apicl(config-dns)# show dns-address
Address           Preferred
-----
192.0.20.123     yes

apicl(config-dns)# no address 192.0.20.123
apicl(config-dns)# show dns-address
Address           Preferred
-----

```

### Executing BASH Commands From the NX-OS Style CLI

To execute a single command in the bash shell, type **bash -c 'path/command'** as shown in this example.

```

apicl# bash -c '/controller/sbin/acidiag avread'

```

You can execute a bash command from any mode or submode in the NX-OS style CLI.

### Entering Configuration Text with Spaces or Special Characters

When a configuration field consists of user-defined text, special characters such as '\$' should be escaped ('\\$\$') or the entire word or string should be wrapped in single quotes to avoid misinterpretation by Bash.

## Differences in Usage from NX-OS

The usage of the NX-OS style CLI for APIC differs from the traditional NX-OS CLI in these ways:

- Global configuration mode is entered with the **configure** command instead of **configure terminal**.
- To perform node-level configuration on a particular leaf switch, you must first navigate to that switch using the **leaf** command.
- The command syntax for specifying a physical port is slightly different. For example, an Ethernet port is specified as **eth x/y** instead of **ethx/y**.
- When a configuration field consists of user-defined text, such as a password, special characters such as '\$' or '!' should be escaped with a backslash ('\\$\$') or the entire word or string should be wrapped in single quotes to avoid misinterpretation by Bash.
- Some command shortcuts are different due to Bash behavior:
  - **Ctrl-D** exits a session.
  - **Ctrl-Z** suspends a job.
- OSPF configuration adds **area route-map** and **area connectivity** commands.

## Limitations with mixing the user interfaces

Because Cisco APIC supports multiple user interfaces (UIs) for configuration, the potential exists for unintended interactions when you create a configuration with one UI and later modify the configuration with another UI. For example, configurations done through the NX-OS-style CLI are rendered in the Cisco APIC GUI. They can be seen, but sometimes might not be editable in the GUI. Similarly, changes made in the Cisco APIC GUI can be seen in the NX-OS-style CLI, but might only partially work.

### Limitation with mixing the NX-OS-style CLI and the Cisco APIC GUI when you configure per-interface

When you configure per-interface, configurations you perform in the Cisco APIC GUI might only partially work in the NX-OS-style CLI.

For example, say that you configure a switch port in the GUI at **Tenants > tenant-name > Application Profiles > application-profile-name > Application EPGs > EPG-name > Static Ports > Deploy Static EPG on PC, VPC, or Interface**. When you use the **show running-config** command in the NX-OS style CLI, you receive output such as:

```
leaf 102
interface ethernet 1/15
switchport trunk allowed vlan 201 tenant t1 application ap1 ep1
exit
exit
```

If you use these commands to configure a static port in the NX-OS style CLI, the following error occurs:

```
apicl(config)# leaf 102
apicl(config-leaf)# interface ethernet 1/15
apicl(config-leaf-if)# switchport trunk allowed vlan 201 tenant t1 application apl ep1 ep1
```

```
No vlan-domain associated to node 102 interface ethernet1/15 encap vlan-201
```

This occurs because the CLI has validations that are not performed by the APIC GUI. For the commands from the **show running-config** command to function in the NX-OS CLI, a vlan-domain must have been previously configured. The order of configuration is not enforced in the GUI.

### Limitation with mixing the user interfaces for the Layer 3 external connectivity configuration modes

This section describes considerations for configuring Layer 3 external connectivity with the NX-OS style CLI when you may also be using other user interfaces.

When you configure Layer 3 external connectivity with the APIC NX-OS style CLI, you have the choice of two modes:

- **Implicit mode:** A simpler mode that is not compatible with the Cisco APIC GUI nor the REST API.
- **Named (or Explicit) mode:** Compatible with the Cisco APIC GUI and the REST API.

In either case, the configuration should be considered read-only in the incompatible UI.

### Differences between the Layer 3 external connectivity configuration modes

In both modes, the configuration settings are defined within an internal container object, the "Layer 3 Outside" (or "L3Out"), which is an instance of the `L3extOut` class in the API. The main difference between the two modes is in the naming of this container object instance:

- **Implicit mode:** The naming of the container is implicit and does not appear in the CLI commands. The CLI creates and maintains these objects internally.
- **Named mode:** You provide the naming of the container. CLI commands in the named mode have an additional `L3Out` field. To configure the named L3Out correctly and avoid faults, you must understand the API object model for external Layer 3 configuration.




---

**Note** Except for the procedures in the Configuring Layer 3 External Connectivity Using the Named Mode section, this guide describes implicit mode procedures.

---

### Guidelines and limitations for the Layer 3 external connectivity configuration modes

The following guidelines and limitations apply to the Layer 3 external connectivity configuration modes:

#### Using implicit mode and named mode together

In the same Cisco APIC instance, you can use both modes together for configuring Layer 3 external connectivity. However, the Layer 3 external connectivity configuration for a given combination of tenant, VRF instance, and leaf switch can be done only through one mode.

#### Use one mode for a tenant VRF instance that is deployed for Layer 3 external connectivity

For a given tenant VRF instance, the policy domain where the external EPG can be placed can be in either the named mode or in the implicit mode. The recommended configuration method is to use only one mode for a given tenant VRF instance combination across all the nodes where the given tenant VRF instance is

deployed for Layer 3 external connectivity. The modes can be different across different tenants or different VRF instances and no restrictions apply.

### Configurations might be validated against inconsistencies

In some cases, an incoming configuration to a Cisco APIC cluster will be validated against inconsistencies, where the validations involve externally-visible configurations (northbound traffic through the L3Outs). An invalid configuration error message will appear for those situations where the configuration is invalid.

### Supported configuration modes for external Layer 3 features

The external Layer 3 features are supported in both configuration modes, with the following exception: route peering and route health injection (RHI) with a Layer 4 to Layer 7 service appliance is supported only in the named mode. Use the named mode across all border leaf switches for the tenant VRF instance where route-peering is involved.

### L3Outs created using the implicit mode CLI are read-only in the GUI

Layer 3 external network objects (also known as an L3Out) created using the implicit mode CLI procedures are identified by names starting with “\_ui\_” and are marked as read-only in the GUI.

### L3Outs created using the implicit mode CLI can become unmodifiable in the CLI if modified using the REST API

The CLI partitions L3Outs by function, such as interfaces, protocols, route map, and EPG. Modifying an L3Out's configuration using the REST API can break this structure, preventing you from modifying the L3Out using the CLI.

To remove the unmodifiable L3Outs, see the Troubleshooting Unwanted \_ui\_ Objects section in the *Cisco APIC Troubleshooting Guide*.

## About the Modes of Configuring Layer 3 External Connectivity

Because APIC supports multiple user interfaces (UIs) for configuration, the potential exists for unintended interactions when you create a configuration with one UI and later modify the configuration with another UI. This section describes considerations for configuring Layer 3 external connectivity with the APIC NX-OS style CLI, when you may also be using other APIC user interfaces.

When you configure Layer 3 external connectivity with the APIC NX-OS style CLI, you have the choice of two modes:

- Implicit mode, a simpler mode, is not compatible with the APIC GUI or the REST API.
- Named (or Explicit) mode is compatible with the APIC GUI and the REST API.

In either case, the configuration should be considered read-only in the incompatible UI.

### How the Modes Differ

In both modes, the configuration settings are defined within an internal container object, the "L3 Outside" (or "L3Out"), which is an instance of the **L3extOut** class in the API. The main difference between the two modes is in the naming of this container object instance:

- Implicit mode—the naming of the container is implicit and does not appear in the CLI commands. The CLI creates and maintains these objects internally.

- Named mode—the naming is provided by the user. CLI commands in the Named Mode have an additional **l3Out** field. To configure the named L3Out correctly and avoid faults, the user is expected to understand the API object model for external Layer 3 configuration.



---

**Note** Except for the procedures in the *Configuring Layer 3 External Connectivity Using the Named Mode* section, this guide describes Implicit mode procedures.

---

### Guidelines and Restrictions

- In the same APIC instance, both modes can be used together for configuring Layer 3 external connectivity with the following restriction: The Layer 3 external connectivity configuration for a given combination of tenant, VRF, and leaf can be done only through one mode.
- For a given tenant VRF, the policy domain where the External-l3 EPG can be placed can be in either the Named mode or in the Implicit mode. The recommended configuration method is to use only one mode for a given tenant VRF combination across all the nodes where the given tenant VRF is deployed for Layer 3 external connectivity. The modes can be different across different tenants or different VRFs and no restrictions apply.
- In some cases, an incoming configuration to a Cisco APIC cluster will be validated against inconsistencies, where the validations involve externally-visible configurations (northbound traffic through the L3Outs). An Invalid Configuration error message will appear for those situations where the configuration is invalid.
- The external Layer 3 features are supported in both configuration modes, with the following exception:
  - Route-peering and Route Health Injection (RHI) with a L4-L7 Service Appliance is supported only in the Named mode. The Named mode should be used across all border leaf switches for the tenant VRF where route-peering is involved.
- Layer 3 external network objects (l3extOut) created using the Implicit mode CLI procedures are identified by names starting with “\_\_ui\_” and are marked as read-only in the GUI. The CLI partitions these external-l3 networks by function, such as interfaces, protocols, route-map, and EPG. Configuration modifications performed through the REST API can break this structure, preventing further modification through the CLI.

For the steps to remove such objects, see *Troubleshooting Unwanted \_ui\_ Objects* in the *APIC Troubleshooting Guide*.