About Cisco ACI/APIC Configuration

Recommended Settings for the Cisco Application Policy Infrastructure Controller

We recommend the following settings for the Cisco Application Policy Infrastructure Controller (Cisco APIC):

Table 1: Recommended Settings for the Cisco APIC

Navigation Path	Property	Value	Description
System > **System Settings** > **Fabric Wide Setting**	Enforce Subnet Check	Put a check in the box.	This feature enforces subnet checks at the VRF instance level, when the Cisco Application Centric Infrastructure (Cisco ACI) learns the IP address as an endpoint from the data plane. Although the subnet check scope is the VRF instance, this feature can be enabled and disabled only globally under the fabric-wide setting policy. You cannot enable this option only in one VRF instance. If you put a check in the box for this option, the fabric will not learn IP addresses from a subnet other than the one configured on the bridge domain. This feature prevents the fabric from learning endpoint information in this scenario.
System > **System Settings** > **Endpoint Controls**	IP Aging Policy	Enabled	The IP aging policy tracks and ages unused IP addresses on an endpoint. Tracking is performed by using the endpoint retention policy, which is configured for the bridge domain to send ARP requests (for IPv4) and neighbor solicitations (for IPv6) at 75% of the local endpoint aging interval. When no response is received from an IP address, that IP address is aged out.

Navigation Path	Property	Value	Description
Fabric > **External Access Policies** > **Policies** > **Global** > **MCP Instance Policy default**	Admin State	Enabled	This enables the Mis-cabling Protocol (MCP)
	Controls: Enable MCP PDU per VLAN	Put a check in the box.	MCP detects other types of loops that can be caused by various issues, such as misconfiguration, that LLDP and STP cannot discover. This option enables MCP to send packets on a per-EPG basis.

About ACI/APIC Interfaces

The single point of management within the Cisco Application Centric Infrastructure (ACI) architecture is known as the Application Policy Infrastructure Controller (APIC). This controller provides access to all configuration, management, monitoring, and health functions. Having a centralized controller with an application programming interface (API) means that all functions configured or accessed through the fabric can be approached through the following interfaces:

- APIC GUI

  The APIC GUI is a browser-based graphical interface to the APIC that communicates internally with the APIC engine by exchanging REST API messages. It includes two modes:

  - Formerly called Advanced Mode, now simply the APIC GUI—Used for large scale configurations, deployments, and operations; enables granular policy controls such as in switch profiles, interface profiles, policy groups, or access entity profiles (AEPs) for automating mass fabric configuration and deployment.
  - Formerly Basic Mode—Up to release 3.1(x), but now removed, this was a simple interface to enable common workflows, the GUI operational mode enables administrators to get started easily with ACI with a minimal knowledge of the object model. The simplified GUI allows the configuration of leaf ports and tenants without the need to configure advanced policies.

  For more information about the APIC GUI, see *Cisco APIC Getting Started Guide, Release 3.x* and *Cisco APIC Basic Configuration Guide, Release 3.x*.

- NX-OS Style CLI—The NX-OS style Command-Line Interface (CLI) can be used for APIC configuration, deployment, and operation. It 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.

  For important guidelines to use both the NX-OS style CLI and the APIC GUI to configure Cisco APIC, see Mixing the NX-OS Style CLI and the APIC GUI, on page 4.

  For more information about the NX-OS style CLI, see *Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*.

- APIC REST API—The REST API is responsible for accepting configuration, as well as providing access to management functions for the controller. This interface is a crucial component for the GUI and CLI, and also provides a touch point for automation tools, provisioning scripts and third party monitoring and management tools.

  The APIC REST API is a programmatic interface that uses REST architecture. The API accepts and returns HTTP (not enabled by default) or HTTPS messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. You can use any programming language to generate the messages and the JSON or XML documents that contain the API methods or MO descriptions.

  For more information about the REST API, see the *Cisco APIC REST API Configuration Guide*.

Mixing the NX-OS Style CLI and the APIC GUI

Basic mode is deprecated since Cisco APIC Release 3.0(1). There is only one GUI as of that release.

Caution Configurations done through the NX-OS style CLI are rendered in the APIC GUI. They can be seen, but sometimes may not be editable in the GUI. Also changes made in the APIC GUI may be seen in the NX-OS style CLI, but may only partially work. See the following examples:

- Do not mix the GUI and the CLI, when doing per-interface configuration on APIC. Configurations performed in the GUI, may only partially work in the NX-OS CLI.

  For example, if 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**

  Then 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 epg ep1
   exit
   exit
  ```

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

  ```
  apic1(config)# leaf 102
  apic1(config-leaf)# interface ethernet 1/15
  apic1(config-leaf-if)# switchport trunk allowed vlan 201 tenant t1 application ap1 epg
   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.

For the steps to remove such objects, see *Troubleshooting Unwanted _ui_ Objects* in the *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*.

Configuration Validation

When the administrator enters a configuration in the Cisco Application Policy Infrastructure Controller (Cisco APIC), the Cisco APIC performs checks to make sure that the configuration is valid, which is known as validation. If the configuration is accepted, but it conflicts with other previous configurations, Cisco APIC or the leaf switches might raise faults. The amount of checks performed by the Cisco APIC before accepting a configuration varies depending on the release. Newer releases have been enhanced to perform more checks before the configuration is accepted instead of only raising faults asynchronously.

The release with the greatest amount of changes in terms of additional validations is the Cisco APIC release 2.3. Cisco APIC release 3.0 further enhances validations at the VRF instance level. As an example, in Cisco APIC release 2.3, for the same VRF instance and the same L3Out, you can define multiple Switch Virtual Interface (SVI) logical interface profiles for the same SVI (encap) with different IP addresses. You can define IP address 10.10.10.1/24 on path node1, port 1/41, VLAN (encap) 10, and IP address 10.10.10.2/24 for path node1, port 1/43, VLAN (encap) 10.

This results in only one IP address being used for SVI 10 on the leaf switch despite the fact that you configured multiple IP addresses, and depending on which IP address is used as the next hop for routing or whether you have IGP configured, the configuration might function properly.

Starting with Cisco APIC release 3.0, the above configuration would not be accepted, because even if in the Cisco Application Centric Infrastructure (Cisco ACI) object model the SVI is defined per path (logical interface profile), a given VRF instance on a given leaf switch can only have one IP address for an SVI and potentially a secondary IP address. Several other validations were also introduced in Cisco APIC release 3.0.

The objective of these validations is to reduce or eliminate configuration errors by informing the user of the errors at the configuration time instead of accepting the configuration and raising faults asynchronously.

As a result of these improvements, if you POST a configuration that was incorrect, but was considered valid prior to the 2.3 release, this POST would not result in the configuration being posted and the Cisco APIC will return an error message.

There might be existing Cisco APIC deployments that are functioning correctly with versions prior to Cisco APIC release 2.3 despite the fact that the configurations might not be valid. To reduce the impact of a firmware upgrade in such scenarios, after you upgrade to the 2.3 release or later, the Cisco APIC relaxes the validation checks on existing configurations.

Cisco APIC also offers the option to import an existing configuration with the "Best Effort" mode instead of the "Atomic" mode. This option offers the ability to accept a configuration even if there are portions that are not valid. The Cisco APIC pushes the valid portions of the configuration and ignores the portions that are not consistent with the validation. For the inconsistent portions, the Cisco APIC issues an error message that is visible when you use the following command:

```
show snapshot jobs import_job
```