Overview of the APIC REST API

This chapter contains the following sections:

About the Application Policy Infrastructure Controller

The Cisco Application Centric Infrastructure (ACI) is a distributed, scalable, multitenant infrastructure with external end-point connectivity controlled and grouped through application-centric policies. The Application Policy Infrastructure Controller (APIC) is the unified point of automation, management, monitoring, and programmability for the ACI. The APIC supports the deployment, management, and monitoring of any application anywhere, with a unified operations model for the physical and virtual components of the infrastructure. The APIC programmatically automates network provisioning and control that is based on the application requirements and policies. It is the central control engine for the broader cloud network; it simplifies management and allows flexibility in how application networks are defined and automated. It also provides northbound Representational State Transfer (REST) APIs. The APIC is a distributed system that is implemented as a cluster of many controller instances.

Management Information Model

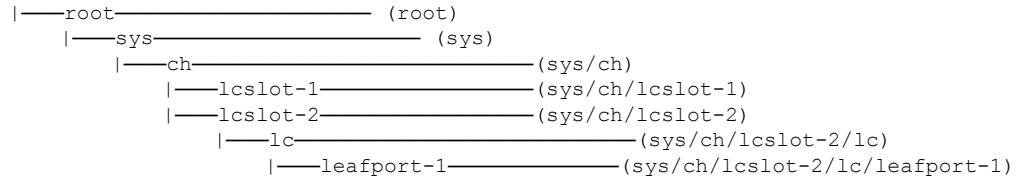
All the physical and logical components that comprise the Application Centric Infrastructure fabric are represented in a hierarchical management information model (MIM), also referred to as the management information tree (MIT). Each node in the tree represents a managed object (MO) or group of objects that contains its administrative state and its operational state.

The hierarchical structure starts at the top (Root) and contains parent and child nodes. Each node in this tree is an MO and each object in the ACI fabric has a unique distinguished name (DN) that describes the object and its place in the tree. MOs are abstractions of the fabric resources. An MO can represent a physical object, such as a switch or adapter, or a logical object, such as a policy or fault.

Configuration policies are the majority of the policies in the system and describe the configurations of different ACI fabric components. Policies determine how the system behaves under specific circumstances. Certain MOs are not created by users but are automatically created by the fabric (for example, power supply objects and fan objects). By invoking the API, you are reading and writing objects to the MIM.

The information model is centrally stored as a logical model by the APIC, while each switch node contains a complete copy as a concrete model. When a user creates a policy in the APIC that represents a configuration, the APIC updates the logical model. The APIC then performs the intermediate step of creating a fully elaborated policy from the user policy and pushes the policy into all the switch nodes where the concrete model is updated. The models are managed by multiple data management engine (DME) processes that run in the fabric. When a user or process initiates an administrative change to a fabric component (for example, when you apply a profile to a switch), the DME first applies that change to the information model and then applies the change to the actual managed endpoint. This approach is called a model-driven framework.

The following branch diagram of a leaf switch port starts at the topRoot of the ACI fabric MIT and shows a hierarchy that comprises a chassis with two line module slots, with a line module in slot 2.



Object Naming

You can identify a specific object by its distinguished name (DN) or by its relative name (RN).

Note You cannot rename an existing object. To simplify references to an object or group of objects, you can assign an alias or a tag.

Distinguished Name

The DN enables you to unambiguously identify a specific target object. The DN consists of a series of RNs:

```
dn = {rn}/{rn}/{rn}/{rn}...
```

In this example, the DN provides a fully qualified path for `fabport-1` from the top of the object tree to the object. The DN specifies the exact managed object on which the API call is operating.

```
< dn ="sys/ch/lcslot-1/lc/fabport-1" />
```

Relative Name

The RN identifies an object from its siblings within the context of its parent object. The DN contains a sequence of RNs.

For example, this DN:

```
<dn = "sys/ch/lcslot-1/lc/fabport-1"/>
```

contains these RNs:

Relative Name	Class	Description
sys	top:System	Top level of this system
ch	eqpt:Ch	Hardware chassis container
lcslot-1	eqpt:LCSlot	Line module slot 1
lc	eqpt:LC	Line (I/O) module
fabport-1	eqpt:FabP	Fabric-facing external I/O port 1

About the APIC REST API

The APIC REST API is a programmatic interface to the Application Policy Infrastructure Controller (APIC) that uses a Representational State Transfer (REST) architecture. The API accepts and returns HTTP 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 managed object (MO) descriptions.

The API model provides major functionality for application development. Configuration and state information of the Cisco Application Centric Infrastructure (ACI) fabric is stored in a hierarchical tree structure known as the management information tree (MIT), which is accessible through the API. You can make changes on a single object or an object subtree. With an API call, you can make changes to the configuration of switches, adapters, policies, and other hardware and software components.

The API operates in forgiving mode, which means that missing attributes are substituted with default values (if applicable) that are maintained in the internal data management engine (DME). The DME validates and rejects incorrect attributes. The API is also atomic. If multiple MOs are being configured (for example, virtual NICs), and any of the MOs cannot be configured, the API stops its operation. It returns the configuration to its prior state, stops the API operation that listens for API requests, and returns an error code.

Updates to MOs and properties conform to the existing object model, which ensures backward compatibility. If existing properties are changed during a product upgrade, they are managed during the database load after the upgrade. New properties are assigned default values.

Full event subscription is enabled. When any MO is created, changed, or deleted because of a user- or system-initiated action, an event is generated. With an API query, you can create a subscription to any future changes in the results of that query.

Operation of the API is transactional and terminates on a single data model. The APIC is responsible for all endpoint communication, such as state updates; users cannot communicate directly to endpoints. In this way, developers are relieved from the task of administering isolated, individual component configurations.

The API model includes these programmatic entities:

- Classes—Templates that define the properties and states of objects in the management information tree.
- Methods—Actions that the API performs on one or more objects.
- Types—Object properties that map values to the object state (for example, equipmentPresence).

A typical request comes into the DME and is placed in the transactor queue in first in, first out (FIFO) order. The transactor gets the request from the queue, interprets the request, and performs an authorization check.

After the request is confirmed, the transactor updates the MIT. This complete operation is done in a single transaction.

Note For detailed reference information about API classes, properties, and data types, see the *Cisco APIC Management Information Model Reference*, which is a web-based application.