



The Cisco Prime Central Information Model and APIs support the following features:

- Managed Elements and Equipment Inventory
- Inventory Object Create/Delete/Update and Attribute Value Change Notifications
- Ethernet Virtual Connection Resource Inventory

The Cisco Prime Central Information Model is designed to conform to the MTOSI 2.0 information model and operational APIs. Cisco reference implementation extensions are introduced to support proprietary APIs and name space as per Cisco Prime release requirements.

The sections below document all the APIs supported by Cisco Prime Central and the reference information of the corresponding WSDL and XSD files. Interfaces are defined in the WSDL and the corresponding data types can be found in XSD files.

## Managed Element Retrieval API

The WSDL document can be obtained at the following URLs:

- [https://<primecentralserver>:<mtosiport>/ManagedElementNamesRetrieval\\_RPC?wsdl](https://<primecentralserver>:<mtosiport>/ManagedElementNamesRetrieval_RPC?wsdl)
- [https://<primecentralserver>:<mtosiport>/ManagedElementRetrieval\\_RPC?wsdl](https://<primecentralserver>:<mtosiport>/ManagedElementRetrieval_RPC?wsdl)

Where *<primecentralserver>* is the hostname or IP address of the Prime Central server and *<mtosiport>* is the port number where the CXF service for MTOSI is running. By default, the port number is 9110.

Below are the supported managed element retrieval APIs:

- `getAllManagedElementNames`
- `getManagedElement`

---

## Equipment Retrieval APIs

The WSDL document can be obtained at the following URL:

[https://<primecentralserver>:<mtosipor>/EquipmentInventoryRetrieval\\_RPC?wsdl](https://<primecentralserver>:<mtosipor>/EquipmentInventoryRetrieval_RPC?wsdl)

Where *<primecentralserver>* is the hostname or IP address of the Prime Central server and *<mtosipor>* is the port number where the CXF service for MTOSI is running. By default, the port number is 9110.

Below are the supported APIs:

- `getAllEquipment`
- `getEquipment`
- `getContainedEquipment`

## Physical Termination Point Retrieval APIs

The WSDL document can be obtained at the following URL:

[https://<primecentralserver>:<mtosipor>/TerminationPointRetrieval\\_RPC?wsdl](https://<primecentralserver>:<mtosipor>/TerminationPointRetrieval_RPC?wsdl)

Where *<primecentralserver>* is the hostname or IP address of the Prime Central server and *<mtosipor>* is the port number where the CXF service for MTOSI is running. By default, the port number is 9110. The following API is supported: `getAllSupportedPhysicalTerminationPoints`

## Physical Inventory Notifications

The Integration Layer (IL) will expose interfaces for enabling clients of the IL to register and receive notifications of interest.

The currently supported IL interfaces consist of the following:

- WS-SOAP interface that supports MTOSI-conformant operations
- JMS interface that supports the same MTOSI operations using serialized Java objects

The IL notification framework should be based on the above interfaces. This entails the following:

- Support the MTOSI-style notification topic registration pattern for the WS-SOAP interface:
- `SubscribeResponse subscribe(`

`Holder<Header> mtopHeader,`

`SubscribeRequest mtopBody`

`) throws SubscribeException;`

- `UnsubscribeResponse unsubscribe(`

`Holder<Header> mtopHeader,`

`UnsubscribeRequest mtopBody`

`) throws UnsubscribeException;`

---

The subscribe request specifies the following:

- Consumer - URI
- Topic - The string indicating the classification of events of interest
- Selector - Notification Filter

The subscribe response contains a subscriptionID.

The unsubscribe request requires the subscriptionID(s) and topic(s) provided at the time of registration.

The Cisco Prime Central information model supports standard JMS topic subscription for JMS interfaces.

On the notification consumer side, each client type (WS-SOAP or JMS) is expected to support a slightly different flow:

WS-SOAP clients must:

- Expose the NotificationConsumer API as defined by the NotificationConsumerHttp.wsdl and documented in the MTOSI R2.0 Framework DDP.
- Use the Subscribe operation to register for notifications.

JMS clients consume POJOs with the following format:

ArrayList<Object> consisting of the following in this order:

1. com.cisco.prime.esb.fmw.api.Header
2. org.tmforum.mtop.fmw.xsd.notmsg.v1.Notify

For Prime Central 1.1, the supported notification types include:

- Object Creation Notification
- Object Deletion Notification
- Attribute Value Change Notification

These notifications are supported across the following objects:

- Managed Element
- Equipment Holder
- Equipment
- Physical Termination Point

Common Inventory is the primary consumer of notifications within the Prime Suite, because it maintains synchronization with the state of live devices. It is anticipated that Common Inventory will consume notifications from the JMS interface. Other consumers are expected to be Fulfillment (JMS) and the Prime Suite NBI users (SOAP).

This table contains the APIs supported for managed elements, equipment inventory and notification of the relevant entities and attributes. This table also lists notification-related APIs on inventory object and attribute value changes.

API	Response	Request	Description
<a href="#">getAllManagedElementNames (...)</a>	<a href="#">getAllManagedElementNamesResponse</a> A list of managed element names.	<a href="#">getAllManagedElementNamesRequest</a>	Returns all managed element names. This is a proprietary API to facilitate the <a href="#">getManagedElement</a> API for a specific managed element retrieval.
<a href="#">getManagedElement (...)</a>	<a href="#">getManagedElementResponse</a> A specific managed element instance.	<a href="#">getManagedElementRequest</a> A list of RDN consisting of MD/ME name-value pairs.	Returns a requested managed element instance. See the corresponding wsdl/xsd files for more details: <ul style="list-style-type: none"> <li><a href="#">ManageResourceInventory/IIS/wsdl/ManagedElementRetrieval/ManagedElementRetrievalHttp.wsdl</a></li> <li><a href="#">ManageResourceInventory/IIS/xsd/ManagedElementRetrievalMessages.xsd</a></li> </ul> <p>Required ME attribute extensions are defined in: <a href="#">NetworkResourceFulfillment/IIS/xsd/Cisco_ME_EQ_Inventory.xsd</a></p>
<a href="#">getAllEquipment(...)</a>	<a href="#">getAllEquipmentResponse</a> A list of all of the equipment and equipment holders contained in a managed element or equipment holders.	<a href="#">getAllEquipmentRequest</a> Name of the ME or equipment holder for which to retrieve contained equipment and equipment holders.	Returns a list of equipment and equipment holders. See the corresponding wsdl/xsd files for more details: <ul style="list-style-type: none"> <li><a href="#">ManageResourceInventory/IIS/wsdl/EquipmentInventoryRetrieval/EquipmentInventoryRetrievalHttp.wsdl</a></li> <li><a href="#">ManageResourceInventory/IIS/xsd/EquipmentInventoryRetrievalMessages.xsd</a></li> </ul> <p>Required equipment attribute extensions are defined in: <a href="#">NetworkResourceFulfillment/IIS/xsd/Cisco_ME_EQ_Inventory.xsd</a></p>

<a href="#">getContainedEquipment(...)</a>	<a href="#">getAllEquipmentResponse</a> A list of all of the equipment and equipment holders contained in the equipment holders.	<a href="#">getAllEquipmentRequest</a> Name of the equipment holder to retrieve the next-level contained equipment and equipment holders.	Returns the next-level contained Equipment Holder and Equipment list. See the corresponding wsdl/xsd files for more details: <ul style="list-style-type: none"> <li>• <a href="#">ManageResourceInventory/IIS/wsdl/EquipmentInventoryRetrieval/EquipmentInventoryRetrievalHttp.wsdl</a></li> <li>• <a href="#">ManageResourceInventory/IIS/xsd/EquipmentInventoryRetrievalMessages.xsd</a></li> </ul> Equipment Attribute Extension as per requirement are defined in: <a href="#">NetworkResourceFulfillment/IIS/xsd/Cisco_ME_EQ_Inventory.xsd</a>
<a href="#">getAllSupportedPhysicalTerminationPoints( ... )</a>	<a href="#">getAllSupportedPhysicalTerminationPointsResponse</a> A list of physical termination endpoints (PTP) of the specified equipment.	<a href="#">getAllSupportedPhysicalTerminationPointsRequest</a> A list of RDNs that specify the containing equipment: MD/ME/EH[EH/EH]/EQ.	Returns a list PTPs. See the corresponding wsdl/xsd files for more details: <ul style="list-style-type: none"> <li>• <a href="#">ManageResourceInventory/IIS/wsdl/TerminationPointRetrieval/TerminationPointRetrievalMessages.wsdl</a></li> <li>• <a href="#">ManageResourceInventory/IIS/xsd/TerminationPointRetrievalMessages.xsd</a></li> </ul> Required PTP attribute extensions are defined in: <a href="#">NetworkResourceFulfillment/IIS/xsd/Cisco_ME_EQ_Inventory.xsd</a>
<a href="#">subscribe(...)</a>	<a href="#">subscribeResponse</a> A unique subscription identifier is returned to the client OS to be used when invoking the unsubscribe() for the given topic. Note that it must uniquely identify the request signature by its three elements: consumerEPR, topic, and selector.	<a href="#">subscribeRequest</a> This operation allows the client to subscribe for notifications.	The subscribe operation used to receive subscriptions from consumers. See the corresponding wsdl/xsd files for more details: <ul style="list-style-type: none"> <li>• <a href="#">Framework/IIS/wsdl/NotificationProducer/NotificationProducerHttp.wsdl</a></li> <li>• <a href="#">Framework/IIS/xsd/NotificationMessages.xsd</a></li> </ul>

<a href="#">unsubscribe(...)</a>	<a href="#">unsubscribeResponse</a> Response message structure of the unsubscribe operation. Note that it is an empty payload. Failure should be handled as an exception.	<a href="#">unsubscribeRequest</a> This operation allows the client to unsubscribe from a previous subscribed notification channel.	The unsubscribe operation used to receive subscription cancellations from consumers. See the corresponding wsdl/xsd files for more details: <ul style="list-style-type: none"> <li>• <a href="#">Framework/IIS/wsdl/NotificationProducer/NotificationProducerHttp.wsdl</a></li> <li>• <a href="#">Framework/IIS/xsd/NotificationMessages.xsd</a></li> </ul>
<a href="#">notify(...)</a>	Not applicable.	Not applicable.	One-way (notification) message structure of the notify operation. This operation allows event notification to a client OS for the following event types: <ul style="list-style-type: none"> <li>• <a href="#">AttributeValueChangeType</a></li> <li>• <a href="#">ObjectCreationType</a></li> <li>• <a href="#">ObjectDeletionType</a></li> </ul> See the corresponding wsdl/xsd files for more details: <a href="#">Framework/IIS/wsdl/NotificationProducer/NotificationProducerHttp.wsdl</a>

## Ethernet Virtual Connection Resource Inventory APIs

The WSDL document can be obtained at the following URL: [https://<primecentralserver>:<mtosipport>/FlowDomainFragmentRetrieval\\_RPC?wsdl](https://<primecentralserver>:<mtosipport>/FlowDomainFragmentRetrieval_RPC?wsdl)

Where *<primecentralserver>* is the hostname or IP address of the Prime Central server and *<mtosipport>* is the port number where the CXF service for MTOSI is running. By default, the port number is 9110.

This WSDL supports the following APIs:

- [getFlowDomainFragment](#)
- [getFlowDomainFragmentRoute](#)
- [getAllFlowDomainFragmentNames](#)

For details about the interfaces and data types, see:

- [ManagedResourceInventory/FlowDomainFragmentRetrieval wsdl file](#)
- [ManagedResourceInventory/FlowDomainFragmentRetrieval xsd file](#)

Shortcut tip: Pressi	Close	Cancel	What are you worki	Enter your status (1	Last update:
Update	Cancel	2097152	en_GB	/s/en/2154/23/_	

The following table shows the APIs to retrieve Ethernet virtual connections (EVCs), plus MTOSI extension points and extensions. Supported service topologies include point-to-point, multipoint-to-multipoint, and point-to-multipoint (E-tree).

API	Function	Extension Point and Extension
<a href="#">getFlowDomainFragment(...)</a>	Retrieves flow domain fragment entities, CTP entities, and the matrix flow domain fragment reference list. If successful, this operation returns EVC, service parameters, EFP (and its parameters), and the forwarding reference list.	<p>Extension points:</p> <ol style="list-style-type: none"> <li>1. FlowDomainFragmentType &gt; LayeredParametersType &gt; LayerRateType</li> <li>2. FlowDomainFragmentType &gt; LayeredParametersType &gt; vendorExtensions</li> <li>3. FlowDomainFragmentType &gt; TerminationPointDataListType &gt; TerminationPointDataType &gt; LayeredParametersType &gt; vendorExtensions</li> </ol> <p>Extensions are defined in:</p> <ul style="list-style-type: none"> <li>• <a href="#">NetworkResourcesBasic/IIS/xsd/LayerRates.xsd</a></li> <li>• <a href="#">NetworkResourcesBasic/IIS/xsd/Cisco_CE_LayeredParam.xsd</a></li> </ul>
<a href="#">getFlowDomainFragmentRoute(...)</a>	Retrieves the matrix flow domain fragment. If successful, this operation returns EoMPLS forwarding entities, including Ethernet pseudowire edge instances or cross-connects, and VFI.	<p>Extension points:</p> <ol style="list-style-type: none"> <li>1. FlowDomainFragmentRouteType -&gt; MatrixFlowDomainFragmentType -&gt; LayeredParameterType -&gt; vendorExtensions</li> <li>2. FlowDomainFragmentRouteType -&gt; MatrixFlowDomainFragmentType -&gt; vendorExtensions</li> </ol> <p>Extensions are defined in:</p> <a href="#">NetworkResourcesBasic/IIS/xsd/Cisco_CE_LayeredParam.xsd</a>

---

<code>getAllFlowDomainFragment(...)</code>	Retrieves names of all the managed flow domain fragments. If successful, this operation returns all Ethernet connection names.	This is a proprietary API introduced to facilitate the client to perform subsequent retrieval of a named Ethernet connection.
--	--	---

---