



# Provisioning CPEs in Promiscuous Mode

---

When a device boots, it requests a configuration from Prime Cable Provisioning and it is this configuration that determines the level of service for the device. During this process, the DHCP server requests the RDU to build a configuration for the device. The RDU generates a configuration and forwards it to all the DPEs that service the provisioning group that the device belongs to. The DPEs can now provide the device with its configuration without going to the RDU.

Prime Cable Provisioning automatically recognizes devices, assigns the appropriate class of service, dynamically creates and generates device configuration files, and activates subscribers. Prime Cable Provisioning provides a single device management platform to support multiple technologies including DOCSIS, and PacketCable.

Prime Cable Provisioning allows service providers to implement either or both of the following workflow models:

- Preprovisioning: Devices are assigned to subscribers and recorded in advance in the provisioning application. When subscribers plug them in, Prime Cable Provisioning automatically assigns the appropriate service level and activates them.
- Autoprovisioning: When subscribers self-register for service, subscriber devices are captured and recorded in the provisioning application. Subscribers are required to register for service before Prime Cable Provisioning configures the device and activates the service.

Device configurations can include customer-required provisioning information such as:

- DHCP IP address selection
- Bandwidth
- Data rates
- Flow control
- Communication speeds
- Level of service

A configuration can contain DHCP configuration and TFTP files for any device. When you install and boot an unprovisioned device, it is assigned a default technology-specific configuration. You can change the default configuration for each technology that Prime Cable Provisioning supports.

The RDU regenerates the configuration for a device when:

- Certain provisioning API calls, such as changing the device Class of Service, are made.

- Validation for a configuration fails. This occurs, for example, when certain parameters of a DHCP request from a device change from initial request parameters.
- A DPE is repopulating its cache.

Every time the RDU regenerates a configuration for a device, the updated configuration is forwarded to the appropriate DPEs.

- [Promiscuous Access for Devices, on page 2](#)

## Promiscuous Access for Devices

This section describes the objects and the properties that are used to control the configuration of devices that are granted promiscuous access.

Devices are said to be given promiscuous access if they are allowed to boot and be configured without being preregistered in Prime Cable Provisioning. Promiscuous access is typically used for devices, such as computers, that appear behind a registered DOCSIS modem. If promiscuous access is not enabled for unknown devices behind a registered DOCSIS modem, the devices receive the default service level.

To grant promiscuous access to a device, you must:

- Enable or disable the promiscuous policy for unknown devices of a given type. Devices for which promiscuous access is enabled are configured according to the policy, instead of receiving the default configuration.
- Specify the Class of Service meant for unknown devices of a given type if the devices are to be given promiscuous access.
- Specify the DHCP Criteria meant for unknown devices of a given type if the devices are to be given promiscuous access.

## Configuring Promiscuous Access

The following table describes the ways in which you can configure a promiscuous policy for a device. The Provisioning group and Technology-specific promiscuous mode configuration are not possible through the Admin UI.

**Table 1: Configuring Promiscuous Access for Devices**

Configuration Scope	Using API Calls...
Provisioning group of Cable Modem—for example, you can configure promiscuous access to allow computers only behind any registered Cable Modem device in a specific provisioning group.	<code>getProvGroupDetails</code> <code>changeProvGroupProperties</code>
Class of Service object of Cable Modem—for example, you can configure promiscuous access to allow computers only behind DOCSIS modems that are associated with a specific Class of Service.	<code>addClassOfService</code> <code>changeClassOfServiceProperties</code> <code>getClassOfServiceProperties</code>

Configuration Scope	Using API Calls...
DHCP Criteria of Cable Modem—For example, you can configure promiscuous access to allow computers only behind DOCSIS modems that are associated with a specific DHCP Criteria.	<i>addDHCPCriteria</i> <i>changeDHCPCriteriaProperties</i> <i>getDHCPCriteriaDetails</i>
Technology-specific defaults—For example, you can configure promiscuous access for computers behind DOCSIS modems using the technology defaults for DOCSIS modems.	<i>changeDefaults</i> <i>getDefaults</i>
System-wide defaults—Global system defaults	<i>changeSystemDefaults</i> <i>getSystemDefaults</i>

## Promiscuous Access and Property Hierarchy

You can configure a promiscuous policy on a number of objects in Prime Cable Provisioning. It is, therefore, important to understand the settings that take precedence. While the policy is configured using properties, the precedence of properties is determined by the Prime Cable Provisioning property hierarchy. The first object in the property hierarchy that has a specific property determines the value that Prime Cable Provisioning is to use.

Prime Cable Provisioning looks up the properties of the promiscuous policy in the property hierarchy of the device's Cable Modem. For example, for a computer, Prime Cable Provisioning looks up the promiscuous policy settings in the property hierarchy of the cable modem, which functions as a relay for the computer. For more details about property hierarchy, see [Property Hierarchy](#). For more details about promiscuous policy see [Properties for Configuring Promiscuous Policy](#).



### Note

When you set the promiscuous policy using technology defaults, the properties must be set on objects associated with the Cable Modem, not the target device type. For example, to enable promiscuous access for computers behind a DOCSIS modem, you can enable promiscuous access on technology defaults for the DOCSIS modem, but not on technology defaults for computers.

The promiscuous policy properties specify the Class of Service, the DHCP Criteria, and whether promiscuous access is enabled or disabled for each device type. If promiscuous mode is enabled for a device, but a search of the device's Cable Modem hierarchy does not locate a match of the Class of Service or DHCP Criteria properties, the default Class of Service or DHCP Criteria for non-promiscuous access are used. For example, if Prime Cable Provisioning is configured to grant promiscuous access to computers, but it cannot locate a promiscuous Class of Service, DHCP Criteria, or both, then it uses the default Class of Service, DHCP Criteria, or both for the computer.

The Class of Service and the DHCP Criteria defaults are configured on the technology defaults of the target device (instead of its Cable Modem) using these properties:

- Class of Service—*/default/classOfService*

The API constant is `TechnologyDefaultsKeys.DEFAULT_CLASS_OF_SERVICE`.

- DHCP Criteria—*/default/dhcpCriteria*

The API constant is `TechnologyDefaultsKeys.DEFAULT_DHCP_CRITERIA`.

## Generating Configurations for Promiscuous Devices

The configuration for promiscuous devices is generated under these conditions:

- The device first appears online and is given promiscuous access.
- An out-of-date DPE is populating its cache and requests configurations for a specific provisioning group.
- Regeneration of the configuration is explicitly requested for the device via the API call *regenConfigs*.
- Configuration of the Cable Modem device for a promiscuous access device is being regenerated.
- Changes to the promiscuous policy (or other configuration changes) prompt the Prime Cable Provisioning Configuration Regeneration Service (CRS) service to regenerate configurations of affected devices.

Every time a configuration for a promiscuous device is regenerated, it uses the newly configured promiscuous policy (for example, the Class of Service currently specified for promiscuous computers). However, if the Class of Service or DHCP Criteria of a device is changed via the API after the device appears online as a promiscuous device, then from then on, the device is not considered promiscuous and is unaffected by any changes that you make to the promiscuous policy. The device is henceforth considered registered.

## Properties for Configuring Promiscuous Policy

To configure promiscuous access for devices, you must configure the properties associated with specific device types that Prime Cable Provisioning supports. You can enable or disable promiscuous access for the device types.

- Enabled—Enables promiscuous access for devices within the scope associated with the API call that [Table 1: Configuring Promiscuous Access for Devices, on page 2](#) describes.
- Disabled—Disables promiscuous access. If the property does not exist, the default is the disabled setting.

See [Table 2: Properties for Enabling Promiscuous Access, on page 5](#) for a list of properties on which you configure promiscuous access.

Promiscuous policy properties are divided into read-write and read-only properties. This section describes the read-write and read-only properties that you must configure to enable promiscuous access for devices and those that you set to select Class of Service or DHCP Criteria for these devices.

### Read-Write Properties



**Note** [Table 1: Configuring Promiscuous Access for Devices, on page 2](#) describes the applicable API calls for all the properties that are described in this section.

[Table 2: Properties for Enabling Promiscuous Access, on page 5](#) describes the properties that you can use to enable promiscuous access.

**Table 2: Properties for Enabling Promiscuous Access**

Property Name	Description
<code>/promiscuousMode/enable/Computer</code>	<p>Sets a Boolean value of “true” or “false” in the Cable Modem’s property hierarchy:</p> <ul style="list-style-type: none"> <li>• true—Enables promiscuous access for computers behind such a relay.</li> <li>• false—Disables promiscuous access for computers behind such a relay.</li> </ul> <p>If the property does not exist in the Cable Modem’s property hierarchy, promiscuous access for computers behind such a relay is disallowed and the devices receive default access.</p>
	<b>API Constant</b> <code>PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED</code>
<code>/promiscuousMode/enable/ PacketCableMTA</code>	<p>Sets a Boolean value of “true” or “false” in the Cable Modem’s property hierarchy:</p> <ul style="list-style-type: none"> <li>• true—Enables promiscuous access for PacketCable MTAs behind such a relay.</li> <li>• false—Disables promiscuous access for PacketCable MTAs behind such a relay.</li> </ul> <p>If the property does not exist in the Cable Modem’s property hierarchy, promiscuous access for PacketCable MTAs behind such a relay is disallowed and the devices receive default access.</p>
	<b>API Constant</b> <code>PolicyKeys.PACKET_CABLE_MTA_PROMISCUOUS_MODE_ENABLED</code>

## Read-Write Properties

Property Name	Description
<i>/promiscuousMode/enable/STB</i>	<p>Sets a Boolean value of “true” or “false” in the Cable Modem’s property hierarchy:</p> <ul style="list-style-type: none"> <li>• true—Enables promiscuous access for STBs behind such a relay.</li> <li>• false—Disables promiscuous access for STBs behind such a relay.</li> </ul> <p>If the property does not exist in the Cable Modem’s property hierarchy, promiscuous access for STBs behind such a relay is disallowed and the devices receive default access.</p>
<i>/promiscuousMode/enable/ CableHomeWanData</i>	<p>Sets a Boolean value of “true” or “false” in the Cable Modem’s property hierarchy:</p> <ul style="list-style-type: none"> <li>• true—Enables promiscuous access for CableHome WAN-Data devices behind such a relay.</li> <li>• false—Disables promiscuous access for CableHome WAN-Data devices behind such a relay.</li> </ul> <p>If the property does not exist in the Cable Modem’s property hierarchy, promiscuous access for WAN-Data devices behind such a relay is disallowed and the devices receive default access.</p>
	<p><b>API Constant</b></p> <p><code>PolicyKeys.STB_PROMISCUOUS_MODE_ENABLED</code></p> <p><code>PolicyKeys.CABLE_HOME_WAN_DATA_PROMISCUOUS_MODE_ENABLED</code></p>

Property Name	Description
<i>/promiscuousMode/enable/ CableHomeWanMan</i>	<p>Sets a Boolean value of “true” or “false” in the Cable Modem's property hierarchy:</p> <ul style="list-style-type: none"> <li>• true—Enables promiscuous access for CableHome WAN-MAN devices behind such a relay.</li> <li>• false—Disables promiscuous access for CableHome WAN-MAN devices behind such a relay.</li> </ul> <p>If the property does not exist in the Cable Modem's property hierarchy, promiscuous access for WAN-MAN devices behind such a relay is disallowed and the devices receive default access.</p>
	<b>API Constant</b> <code>PolicyKeys.CABLE_HOME_WAN_MAN_PROMISCUOUS_MODE_ENABLED</code>
<i>/promiscuousMode/enable/ERouter</i>	<p>Sets a Boolean value of “true” or “false” in the Cable Modem's property hierarchy:</p> <ul style="list-style-type: none"> <li>• true—Enables promiscuous access for eRouters behind such a relay.</li> <li>• false—Disables promiscuous access for eRouters behind such a relay.</li> </ul> <p>If the property does not exist in the Cable Modem's property hierarchy, promiscuous access for eRouters behind such a relay is disallowed and the devices receive default access.</p>
	<b>API Constant</b> <code>PolicyKeys.EROUTER_PROMISCUOUS_MODE_ENABLED</code>
<i>/promiscuousMode/enable/</i>	<p>Use this property to enable or disable promiscuous access for new types of devices by appending the property name with the name of a valid device type.</p> <p>Sets a Boolean value of “true” or “false.”</p>
	<b>API Constant</b>
	<code>PolicyKeys.PROMISCUOUS_MODE_PREFIX</code>

The following table describes the read-write properties that you must configure to select Class of Service for devices granted promiscuous access.

**Table 3: Promiscuous Access—Read-Write Properties for Class of Service**

<b>Class of Service Property Name</b>	<b>Description</b>
/provisioning/cpeClassOfService/Computer	Specifies the name of an existing Class of Service that will be selected for promiscuous computers  <b>API Constant</b> PolicyKeys.COMPUTER_CLASS_OF_SERVICE
/provisioning/cpeClassOfService/PacketCableMTA	Specifies the name of an existing Class of Service that will be selected for promiscuous PacketCable MTAs  <b>API Constant</b> PolicyKeys.PACKET_CABLE_MTA_CLASS_OF_SERVICE
/provisioning/cpeClassOfService/STB	Specifies the name of an existing Class of Service that will be selected for promiscuous set-top boxes  <b>API Constant</b> PolicyKeys.STB_CLASS_OF_SERVICE
/provisioning/cpeClassOfService/CableHomeWanMan	Specifies the name of an existing Class of Service that will be selected for promiscuous CableHome WAN-Data devices  <b>API Constant</b> PolicyKeys.CABLEHOME_WAN_DATA_CLASS_OF_SERVICE
/provisioning/cpeClassOfService/CableHomeWanData	Specifies the name of an existing Class of Service that will be selected for promiscuous CableHome WAN-MAN devices  <b>API Constant</b> PolicyKeys.CABLEHOME_WAN_MAN_CLASS_OF_SERVICE
/provisioning/cpeClassOfService/ERouter	Specifies the name of an existing Class of Service that will be selected for promiscuous eRouters  <b>API Constant</b> PolicyKeys.EROUTER_CLASS_OF_SERVICE
/provisioning/cpeClassOfService/	Specifies an existing Class of Service that will be selected for devices of the specified device type. Use this property name with a valid device type name. You can use this property for custom device types.  <b>API Constant</b> PolicyKeys.PROMISCUOUS_COS_PREFIX

The following table describes the read-write properties that you must configure to select DHCP Criteria for devices granted promiscuous access.

**Table 4: Promiscuous Access—Read-Write Properties for DHCP Criteria**

DHCP Criteria Property Name	Description
/provisioning/cpeDhcpCriteria/ Computer	Specifies the name of an existing DHCP Criteria object that will be selected for promiscuous computers  <b>API Constant</b> <code>PolicyKeys.COMPUTER_DHCP_CRITERIA</code>
/provisioning/cpeDhcpCriteria/ PacketCableMTA	Specifies the name of an existing DHCP Criteria object that will be selected for promiscuous PacketCable MTAs  <b>API Constant</b> <code>PolicyKeys.PACKET_CABLE_MTA_DHCP_CRITERIA</code>
/provisioning/cpeDhcpCriteria/STB	Specifies the name of an existing DHCP Criteria object that will be selected for promiscuous set-top boxes  <b>API Constant</b> <code>PolicyKeys.STB_DHCP_CRITERIA</code>
/provisioning/cpeDhcpCriteria/ CableHomeWanData	Specifies the name of an existing DHCP Criteria object that will be selected for promiscuous CableHome WAN-Data devices  <b>API Constant</b> <code>PolicyKeys.CABLEHOME_WAN_DATA_DHCP_CRITERIA</code>
/provisioning/cpeDhcpCriteria/ CableHomeWanMan	Specifies the name of an existing DHCP Criteria object that will be selected for promiscuous CableHome WAN-MAN devices  <b>API Constant</b> <code>PolicyKeys.CABLEHOME_WAN_MAN_DHCP_CRITERIA</code>
/provisioning/cpeDhcpCriteria/ERouter	Specifies the name of an existing DHCP Criteria object that will be selected for promiscuous eRouters  <b>API Constant</b> <code>PolicyKeys.EROUTER_DHCP_CRITERIA</code>
/provisioning/cpeDhcpCriteria/	Specifies an existing DHCP Criteria object that will be selected for devices of the specified device type. Use this property name with a valid device type name. You can use this property for custom device types.  <b>API Constant</b> <code>PolicyKeys.PROMISCUOUS_DC_PREFIX</code>

## Read-Only Properties

### Read-Only Properties

The following table covers read-only promiscuous properties that you must configure to select the Class of Service and the DHCP Criteria for devices. Together with the read-write properties specified in the previous section, these read-only properties help determine the current system configuration.

**Table 5: Promiscuous Access—Read-Only Properties**

Property Name	Description	
<i>/isSystemWide/default/promiscuous</i>	Returns a “true” value if a given Class Of Service or DHCP Criteria object is referenced as system-wide default for promiscuous devices.	
	<b>Applicable API Calls</b> <i>getClassOfServiceProperties</i> <i>getDHCPCriteriaDetails</i>	<b>API Constant</b> PolicyKeys.IS_SYSTEM_WIDE_DEFAULT_PROMISCUOUS
<i>/referencedBy/deviceTypes/forPromiscuousDevices</i>	Returns a list of Device Type object (technology) names that reference a given Class of Service or DHCP Criteria object in promiscuous policy properties	
	<b>Applicable API Calls</b> <i>getClassOfServiceProperties</i> <i>getDHCPCriteriaDetails</i>	<b>API Constant</b> PolicyKeys.REFERENCED_BY_DEVICE_TYPE_FOR_PROMISCUOUS_DEVICES
<i>/related/classesOfService</i>	Returns a list of Class of Service object names that are used by a given Class of Service or DHCP Criteria object in promiscuous policy properties  <b>Note</b> You can use this property as a shortcut to obtain the Class of Service list. You can also obtain this list by reading individual promiscuous policy properties set on this object.	
	<b>Applicable API Calls</b> <i>getClassOfServiceProperties</i> <i>getDHCPCriteriaDetails</i>	<b>API Constant</b> PolicyKeys.RELATED_CLASS_OF_SERVICE
<i>/related/dhcpCriteria</i>	Returns a list of DHCP Criteria object names that are used by a given Class of Service or DHCP Criteria object in promiscuous policy properties  <b>Note</b> You can use this property as a shortcut to obtain the DHCP Criteria list. You can also obtain this list by reading individual promiscuous policy properties set on this object.	
	<b>Applicable API Calls</b> <i>getClassOfServiceProperties</i> <i>getDHCPCriteriaDetails</i>	<b>API Constant</b> PolicyKeys.RELATED_DHCP_CRITERIA

## Custom Policy for Promiscuous Devices

You can configure promiscuous policy for a device using the properties specified in the above section. When additional logic is required, however, you can implement custom logic using extensions and custom properties. Custom properties allow for the definition of new properties, which can then be stored on any object via the API.

To augment the promiscuous device policy, you can use these extensions:

- Device Detection—Determines the technology type of the device (usually based on DHCP request data). Information that this extension detects is placed in a Device Detection Context that other extensions then use.
- Service-Level Selection—Selects the appropriate Class of Service and DHCP Criteria objects for a device. The promiscuous policy properties determine the Class of Service and DHCP Criteria for devices with promiscuous access.
- Configuration Generation—Generates the configuration for a device and, if necessary, for the devices behind it. Configurations are regenerated for promiscuous devices behind the Cable Modem based on the policy that the service-level selection extension selects. You may need to change the extension only if you want to augment the default behavior of regenerating configuration for devices behind a Cable Modem.

