Frequently Asked Questions

This chapter provides a list of frequently asked questions about Prime Cable Provisioning.

- Prime Cable Provisioning Configuration, on page 1
- IPv6 Configuration, on page 4
- CMTS Configuration, on page 7
- Custom Relay Agent Remote ID Validation for RPD Devices, on page 9

Prime Cable Provisioning Configuration

This section features FAQs related to general Prime Cable Provisioning configurations.

- How do I enable or disable Network Registrar extensions?
- How do I enable tracing for Network Registrar extensions?
- Why does the DPE server registration fails?
- Why does RDU crash while updating the agent.conf?
- Why are the components not been able to communicate?
- Why does execution of a reliable batch fails for Radius-only user?
- Why does BAC 4.x and 4.x.x API clients get unrecognized batch ID?
- How do I attach External Agent into PCP Components?, on page 4

How do I enable or disable Network Registrar extensions?

The procedures described in this section assume that:

- The Prime Cable Provisioning component is installed in */opt/CSCObac*.
- Cisco Prime Network Registrar is installed in */opt/nwreg2*.

Manually install Network Registrar extension points

To manually install Network Registrar extension points:

Step 1 Log into the Network Registrar server, with *root* access.

Step 2 Take a backup and copy the *libbprextensions.so* directory to the *NR_HOME/local/extensions/dhcp/dex/* directory.

Step 3 Take a backup and copy the *cnr_ep.properties* file to the *BPR_HOME/cnr_ep/conf* directory.

Step 4 Configure extensions from the Network Registrar command-line tool (**nrcmd**) using:

NR_HOME/**local/usrbin/nrcmd -s -b <** *BPR_HOME*/**cnr_ep/bin/bpr_cnr_enable_extpts.nrcmd**

Step 5 Reload the DHCP server.

Manually disable Network Registrar extension points

To manually disable Network Registrar extension points:

Step 1 Log into the Network Registrar server, with *root* access.

Step 2 Enter:

NR_HOME/**local/usrbin/nrcmd -s -b <** *BPR_HOME*/**cnr_ep/bin/bpr_cnr_disable_extpts.nrcmd**

Step 3 Delete the *libbprextensions.so* file, which is located in the *NR_HOME/local/extensions/dhcp/dex/* directory.

Step 4 Reload the DHCP server.

How do I enable tracing for Network Registrar extensions?

To enable tracing for Network Registrar extension points:

Step 1 Log into the Network Registrar web UI. The default login and password are **admin** and **changeme**.

Step 2 From the menu, click **DHCP > DHCP Server** page.

The Manage DHCP Server page appears.

Step 3 Click the DHCP Server link.

The Edit DHCP Server page appears.

Step 4 Expand the Extensions category, and set the **extension-trace-level** value as 3 or 4.

Step 5 To view incoming and outgoing packets, expand the Logging category, and select the **incoming-packet-detail** and **outgoing-packet-detail** check boxes.

Step 6 Click **Modify Server**.

Step 7 Reload the DHCP server.

Why does the DPE server registration fails?

The registration of your DPE servers may be failing because the DPEs are not up to the requirements of the provisioning group.

Check the DPE log files for error messages that indicate that you must:

- Enable additional configuration, for example, if you must enable the TFTP service on the DPE.
- Upgrade the servers to enable features that are available only in Prime Cable Provisioning.
- Upgrade all the extension points before enabling IPv6 PG communication.

Why does RDU crash while updating the agent.conf?

For RDU or DPE, you must not configure any extended JVM arguments through agent.conf.

Why are the components not been able to communicate?

The firewall must be disabled on the servers on which the Prime Cable Provisioning components are installed. To know the ports that are being used by Prime Cable Provisioning, see Port Information from the Support Site.

Why does connection drop between a legacy Solaris DPE and Linux RDU

Drop in connection between Solaris DPE and Linux RDU occurs when there is a huge regeneration of device configuration behind a single CM or when configuration regeneration for 1 million CM devices happens using CoS level change where a CRS job is kicked in.

Why does execution of a reliable batch fails for Radius-only user?

A reliable batch submitted by a Radius only user cannot be guaranteed to execute across reboots or when the user logs out. This applies to those users that do not have their privileges defined in an RDU account but are only provided by a remote Radius server.

Why does BAC 4.x and 4.x.x API clients get unrecognized batch ID?

In Prime Cable Provisioning 5.0, a few new command error messages as well as error codes are added that the API clients of earlier releases do not recognize. This results in the API clients asking for the batch response to the RDU. But as the batch was already processed and the response was sent earlier, the RDU does not remember the batch response and hence responds with an error message from the second attempt onwards.

For successful communication between the Prime Cable Provisioning RDU and 4.x and 4.x.x API clients, ensure that the 5.0 bpr.jar, bacbase.jar, and bac-common.jar files are copied to the 4.x and 4.x.x API client setup. These jars are loaded to the appropriate classpaths. Cisco recommends that you use Java version 1.6.0_32 or later to support the API client in Prime Cable Provisioning.

Note Note: If you are upgrading from 4.2.x to 5.0, ensure that all your API clients are upgraded to 4.2.1 before upgrading the RDU to 5.0. After upgrading all the RDUs to 5.0 upgrade all the API clients to 5.0.

Some of the scenarios in which the RDU can send new command error code are listed below.

- User does not have the specific privileges to execute a given command.
- Instance level authorization failure.
- User information (accessible privileges, accessible domains and active sessions) is not cached in the RDU. This can happen for a persistent API client with Prime Cable Provisioning jars prior to 4.2, as they do not re-authenticate while reconnecting to the RDU. Hence the user information is not cached in the RDU. The workaround to resolve this issue is to restart the API client after every RDU restart and the user must re-authenticate after the restart of the API client. To fix the problem upgrade API client jars of earlier releases to Prime Cable Provisioning 5.0 release.

Note The user information is stored in the RDU cache only when the user is authenticated . The user information remains in the cache until the last active user session expires or is terminated.

Why does the Split Brain of the filesystem occur?

Split Brain of the filesystem occurs if auto-failback and auto split-brain is set to *No* during the RDU HA installation, and if both primary and secondary servers come online at the same time. In this case, you must manually correct the split brain using utility scripts.

How do I attach External Agent into PCP Components?

The following JVM standard options are supported by the PCP components: RDU, DPE CLI, DPE, KDC, and SNMP Agent.

Option	Description
- agentlib:< libname>[=< options>]	To load native agent library <libname>.
-agentpath:<pathname>[=<options>]	To load native agent library by full pathname.
-javaagent:<jarpath>[=<options>]	To load Java programming language agent.

Using these options, you can attach any external agent into the PCP component and monitor the JVM performance.

IPv6 Configuration

This section features FAQs related to IPv6 while configuring Prime Cable Provisioning.

- How do I enable provisioning in IPv6 for DPE?

- How do I configure an IPv4 interface for provisioning?
- DPE is configured for IPv6 provisioning, but Prime Cable Provisioning does not provision IPv6 DOCSIS 3.0 devices. Why?
- When searching for all devices using their MAC address, some IPv6 devices do not show up. Why?
- How do I enable IPv6 on an interface?
- How do I configure IPv6 on a loopback interface?
- How do I assign a static IP address to an interface?

How do I enable provisioning in IPv6 for DPE?

To enable IPv6 provisioning for the DPE, complete this procedure from the DPE command line:

Step 1 For enabling IPv6 provisioning, you must configure two interfaces using the following commands:

a) To configure the DPE to use the specified interface, identified by its IP address, when communicating with Network Registrar extensions, enter:

interface ip *ipv4_address* **pg-communication**

ipv4_address—Identifies the IPv4 address of a specific DPE interface.

interface ip *ipv6_address* **pg-communication**

ipv6_address—Identifies the IPv6 address of a specific DPE interface.

Note
- You can configure either IPv4 address only, or both IPv4 and IPv6 addresses by using this command.
- If you configure an interface(IPv4 / IPv6) to communicate with the extensions (using **interface ip pg-communication** command), the extensions communicate with the DPE via the interface you specify.
- If only the IPv4 address is specified, the interface for communication with Network Registrar extensions, the extensions communicate with DPE via the specified IPv4 interface for both IPv4 and IPv6 mode.
- If both IPv4 and IPv6 addresses are specified the interfaces for communication with Network Registrar extensions, the extensions communicate with DPE via the specified IPv4 interface in case of IPv4 mode, and the specified IPv6 interface in case of IPv6 mode.
- IPv6 global address or link local address can be used in the **interface ip pg-communication** command.
- Using this configuration, you can enable the use of split-networking techniques to isolate devices facing communication from management communications.
- If you do not specify any interface for communication with Network Registrar extensions, the extensions communicate with the DPE via the interface on which provisioning is enabled.

b) To configure the specified interface, identified by its IP address, to handle provisioning requests, enter:

interface ip *ip_address* **provisioning**

ip_address—Specifies the IP address of the interface in the IPv6 format.

Step 2 Enable these services using the respective commands:

- TFTP—**service tftp** *1..1* **ipv6 enabled true**
- ToD—**service tod** *1..1* **ipv6 enabled true**

Step 3 Reload the DPE using the **dpe reload** command.

How do I configure an IPv4 interface for provisioning?

To configure an IPv4 interface for provisioning, you must set the fully qualified domain name (FQDN) for that interface using this command:

interface ip *ip_address* **provisioning fqdn** *fqdn*

- *ip_address*—Specifies the IP address of the interface in the IPv4 format.
- *fqdn*—Identifies the FQDN that is set on the specified interface.

DPE is configured for IPv6 provisioning, but Prime Cable Provisioning does not provision IPv6 DOCSIS 3.0 devices. Why?

You must enable DOCSIS 3.0 for the provisioning group to which the DPE belongs.

On the Prime Cable Provisioning Admin UI:

Step 1 Choose **Servers > Provisioning Group**.

The Provisioning Group Details page appears.

Step 2 Click the Provisioning Groups link corresponding to the specific DPE.

Step 3 In the Capabilities Management area, click the **Enabled** radio button corresponding to IPv6 - DOCSIS 3.0.

Step 4 Click **Submit**.

When searching for all devices using their MAC address, some IPv6 devices do not show up. Why?

Some IPv6 devices do not appear following a search for all devices using the MAC address option because devices such as the Vista IPv6 computer do not report their MAC address in the Solicit message. As a result, they are known only by their DUID.

If a device reports its MAC address in the CableLabs Device ID option, then you can locate that device using its DUID or its MAC address.

How do I enable IPv6 on an interface?

To enable IPv6 on an interface, run the following commands:

```
# ifconfig intf inet6 plumb up
# /usr/lib/inet/in.ndpd
# touch /etc/hostname6.intf
```

where *intf* identifies the interface on which you want to enable IPv6.

How do I configure IPv6 on a loopback interface?

Before you configure IPv6 on a loopback interface, confirm if the loopback interface is up using this command:

```
# ifconfig -a
```

If the loopback interface is not up, log in as *root* and run the following commands:

```
# ifconfig lo0 inet6 plumb
# route add -inet6 ::1/128 localhost
# ifconfig lo0 inet6 up
```

How do I assign a static IP address to an interface?

While assigning a static IP address is not essential, to do so, run this command:

```
# ifconfig bge0 inet6 addif 2001:420:3800:601::1/64 up
```

CMTS Configuration

This section describes some FAQs related to configuring a cable modem termination system (CMTS):

- How do I know that both cable line cards are using the cable bundle 1?
- Is there an IPv6 cable-helper address that I can use?
- How do I configure multiple IPv6 subnets similar to IPv4 primary and secondary IPv4 subnets?
- How do I view the list of IPv6 modems on the CMTS?
- How do I configure a CMTS interface to accept only IPv6 single stack?
- What does the modem state init(x) mean?

How do I know that both cable line cards are using the cable bundle 1?

You must add this setting for each cable interface:

```
interface Cable3/0
 cable bundle 1
```

Is there an IPv6 cable-helper address that I can use?

Yes, this setting on the bundle is equivalent to the helper-address in IPv4:

```
ipv6 dhcp relay destination FC00:420:3800:710::2 GigabitEthernet0/1
```

How do I configure multiple IPv6 subnets similar to IPv4 primary and secondary IPv4 subnets?

While you can assign multiple prefixes to a bundle for IPv6, there are no primary or secondary types for these subnets in IPv6.

How do I view the list of IPv6 modems on the CMTS?

Use the following command to see the list of IPv6 modems:

```
show cable modem ipv6
```

How do I configure a CMTS interface to accept only IPv6 single stack?

You must add this option to the interface of the cable modem termination system (CMTS):

```
(config-if)# cable ip-init ipv6
```

What does the modem state init(x) mean?

The **show cable modems** (scm) command displays the connected cable modems and their respective states.

The following table lists the various modem states in both IPv4 and IPv6.

Table 1: Cable Modem States

State	Description
IPv4	
init(d)	DHCP Discover
init(io)	DHCP Offer
init(dr)	DHCP Request
init(i)	DHCP Ack
init(o)	TFTP Request

State	Description
Init(t)	ToD Request
online	Online
IPv6	
init6(s)	Solicit
init6(a)	Advertise
init6(r)	Request
init6(i)	Reply
init6(o)	IPv6 TFTP Request
init6(t)	IPv6 ToD request
online	Online

Custom Relay Agent Remote ID Validation for RPD Devices

The relay agent remote ID DHCP option, *relay-agent-remote-id*, is mandatory for DHCPv4 devices. If this option is not present in the request, then the CNR-EP will show an error and drop the packet.

For RPD devices, if you have to support RPD devices without *relay-agent-remote-id*, then the validation has to be moved to RDU device detection extension by using the following steps:

1. 1. Change to NR defaults -> Attributes Required In DHCPv4 Request Dictionary, change the attribute value *relay-agent-remote-id* to *chaddr*. Reload the dhcp, then *relay-agent-remote-id* will become non-mandatory and *chaddr* as mandatory at CNR.
1. 2. Configure the remote id validation custom device detector extensions along with the default device detectors.

```
com.cisco.provisioning.cpe.extensions.builtin.detection.EnableRemoteIdPresenceValidation,com.cisco.provisioning.cpe.extensions.
builtin.detection.DeviceDetector,com.cisco.provisioning.cpe.extensions.builtin.detection.RemoteIdOptionDetector
```

These two custom device detection extensions are already available in bpr.jar.

Now the *relay-agent-remote-id* validation will happen at extension level and only RPD devices are allowed without *relay-agent-remote-id*, but for the other type of devices, detection will throw error if *relay-agent-remote-id* value is not found.