



Configuring Prime Cable Provisioning Technologies

This chapter describes the tasks that you must perform when configuring Prime Cable Provisioning to support specific technologies:

- [Configuring DOCSIS, page 1](#)
- [Configuring PacketCable, page 5](#)
- [Configuring DPoE, page 24](#)
- [Configuring CableHome, page 25](#)

Configuring DOCSIS

This section describes the tasks that you must perform when configuring Prime Cable Provisioning to support the DOCSIS technologies.



Note

See [Technology Option Support](#), for information on DOCSIS options supported by this Prime Cable Provisioning release.

DOCSIS Workflow

Prime Cable Provisioning supports these versions of the DOCSIS specifications: 1.0, 1.1, 2.0, and 3.0.

To successfully configure Prime Cable Provisioning for DOCSIS operations, you must configure the components as described in [Configuring Prime Cable Provisioning Components](#), in addition to those described in this section.

The following table identifies the workflow to follow when configuring Prime Cable Provisioning to support DOCSIS.

Table 1: DOCSIS Workflow

	Task	Refer to...
Step 1	Configure the RDU	
	a. Configure all provisioned DHCP Criteria.	Configuring DHCP Criteria
	b. Configure provisioned Class of Service.	Configuring Class of Service
	c. Configure the promiscuous mode of operation.	System Defaults
Step 2	Configure the DPE	
	a. Enable the TFTP service.	The service tftp /.. / ipv4 ipv6 enabled true command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide .
	b. Optionally, enable the ToD service.	The service tod /.. / ipv4 ipv6 enabled true command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide .
Step 3	Configure Cisco Prime Network Registrar	
	Configure client classes/selection tags to match those added for the provisioned DOCSIS modem DHCP Criteria.	Cisco Prime Network Registrar 8.1 User Guide

DOCSIS Shared Secret

Prime Cable Provisioning lets you define a different DOCSIS shared secret (DSS) for each cable modem termination system (CMTS). In this way, a compromised shared secret affects only a limited number of CMTS, instead of every CMTS in the deployment.

Although the DSS can be set for each DPE, you should set it on a provisioning-group basis. Also, ensure that it matches what has been configured for the CMTS in that provisioning group.



Caution

Configuring multiple DSS within one provisioning group could, under some conditions, result in degraded CMTS performance. However, this factor has virtually no effect on Prime Cable Provisioning.

You can enter the shared secret as a clear text string or as an IOS-encrypted string. When entered in clear text, the DSS is encrypted to suit IOS version 12.2BC.

You can also set the DSS from the RDU using the Admin UI or the API. In this case, the DSS is entered, stored at the RDU, and passed to all DPEs in clear text. Consequently, before a DSS entered this way is stored on the DPE, it is encrypted.

If you set the DSS directly at the DPE using the **dpe docsis shared-secret** command from the CLI, this DSS takes precedence over the one set from the RDU.

Resetting the DOCSIS Shared Secret

You can reset the DSS if the security of the DSS is compromised or to simply change the shared secret for administrative purposes.

To reset the DSS, run the **show running-config** command from the CMTS CLI, then copy and paste the DOCSIS shared secret from the configuration that appears into the DPE configuration. In this way, you can copy the configuration that you enter in a Cisco CMTS into the DPE CLI.



Note

To change the shared secret as described, the CMTS must be running a software version later than version 12.2BC.



Note

For details about the commands mentioned above, and the specific security privileges to run these commands, see the [Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide](#).

To change the DSS:

-
- Step 1** Identify the provisioning group on which you need to reset the DOCSIS shared secret.
 - Step 2** Examine the list of DPEs and CMTS associated with the provisioning group.
 - Step 3** Change the primary DSS on the CMTS.
 - Step 4** Change the compromised DSS on the CMTS to the secondary DSS. This change is required to allow cable modems to continue to register until all the DOCSIS configuration files are successfully changed to use the new DSS.
 - Step 5** Determine which DPEs were affected and change the DSS on each accordingly.
 - Step 6** Confirm that the DOCSIS configuration files are using the new DSS and then remove the compromised secondary shared secret from the CMTS configuration.
-

Extended CMTS MIC Shared Secret

Prime Cable Provisioning lets you define a different Extended CMTS MIC (EMIC) shared secret for each cable modem termination system (CMTS) for EMIC calculation.

The CMTS must support a configuration for the shared secret for EMIC calculation to differ from the shared secret for pre-3.0 DOCSIS CMTS MIC calculation. In the absence of such configuration, the CMTS **MUST** use the same shared secret for Extended CMTS MIC Digest calculation as for pre-3.0 DOCSIS CMTS MIC digest calculation.

In this way, a compromised shared secret affects only a limited number of CMTS, instead of every CMTS in the deployment.

Similar to DSS, EMIC DOCSIS shared secret can be set for each DPE, you should set it on a provisioning-group basis. Also, ensure that it matches what has been configured for the CMTS in that provisioning group.

**Caution**

Configuring multiple EMIC DOCSIS Shared Secret within one provisioning group could, under some conditions, result in degraded CMTS performance. However, this factor has virtually no effect on Prime Cable Provisioning.

You can enter the shared secret as a clear text string or as an IOS-encrypted string. When entered in clear text, the EMIC shared secret is encrypted to suit IOS version 12.2BC.

You can also set the EMIC Shared Secret from the RDU using the Admin UI or the API. In this case, the DOCSIS shared secret is entered, stored at the RDU, and passed to all DPEs in clear text. Consequently, before an Extended MIC shared secret entered this way is stored on the DPE, it is encrypted.

If you set the Extended MIC shared secret directly at the DPE using the **dpe docsis emic shared-secret** command from the CLI, this Extended MIC shared secret takes precedence over the one set from the RDU.

Resetting the Extended EMIC Shared Secret

You can reset the Extended MIC shared secret if the security of the EMIC shared secret is compromised or to simply change the shared secret for administrative purposes.

To reset the DSS, run the **show running-config** command from the CMTS CLI, then copy and paste the EMIC shared secret from the configuration that appears into the DPE configuration. In this way, you can copy the configuration that you enter in a Cisco CMTS into the DPE CLI.

**Note**

To change the shared secret as described, the CMTS must be running a software version later than version 12.2(11)CX.

**Note**

For details about the commands mentioned above, and the specific security privileges to run these commands, see the [Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide](#).

To change the Extended MIC shared secret:

-
- Step 1** Identify the provisioning group on which you need to reset the EMIC shared secret.
 - Step 2** Examine the list of DPEs and CMTS associated with the provisioning group.
 - Step 3** Change the primary EMIC shared secret on the CMTS.
 - Step 4** Change the compromised EMIC shared secret on the CMTS to the secondary EMIC shared secret. This change is required to allow cable modems to continue to register until all the DOCSIS configuration files are successfully changed to use the new DSS.
 - Step 5** Determine which DPEs were affected and change the EMIC shared secret on each accordingly.
 - Step 6** Confirm that the DOCSIS configuration files are using the new EMIC shared secret and then remove the compromised secondary shared secret from the CMTS configuration.
-

Configuring PacketCable

This section describes the configuration of Prime Cable Provisioning to support the PacketCable technologies and bring a PacketCable voice deployment into service.

PacketCable 2.0 supports the convergence of voice, video, data, and mobility technologies. It is based on Session Initiation Protocol (SIP) and IP multimedia system (IMS) and supports configuration and management of Non-Embedded User Equipment (UE) as well as Embedded User Equipment (E-UE).

Prime Cable Provisioning supports only the UEs that are embedded with a DOCSIS Cable Modem and are called as E-UE or Embedded Digital Voice Adapter (E-DVA). E-DVA supports RST (Residential SIP Telephony).

Prime Cable Provisioning supports E-DVA provisioning in IPv4 mode in both PacketCable Basic and Secure modes, and E-DVA provisioning in IPv6 mode only in PacketCable Basic mode.

This section contains information on these variants of PacketCable:

- [Configuring PacketCable Basic, on page 7](#)
- [Configuring PacketCable Secure, on page 11](#)

For information that will help you solve issues in a PacketCable voice technology deployment, see [Troubleshooting PacketCable Provisioning](#).

This chapter assumes that you are familiar with the contents of the PacketCable Multimedia Terminal Adapter (MTA) Device Provisioning Specification, PKT-SP-PROV1.5-I03-070412. For details, see the PacketCable website.

PacketCable Workflows

Prime Cable Provisioning supports these versions of the PacketCable specifications: 1.0, 1.5 and 2.0.

Prime Cable Provisioning also supports two variants of PacketCable voice services: the default Secure mode and the non-secure Basic mode. PacketCable Basic is much the same as the standard PacketCable, except for the lack of security found in the non-secure variant.

This section identifies the tasks that you must perform for each variant.

- [PacketCable Basic, on page 5](#)
- [PacketCable Secure, on page 9](#)

**Note**

The workflows in this section assume that you have loaded an appropriate PacketCable configuration file and the correct MIBs.

PacketCable Basic

You perform the PacketCable-related tasks described in this section only after completing those described in [Configuring Prime Cable Provisioning Components](#).

The following table identifies the workflow to follow when configuring PacketCable Basic on Prime Cable Provisioning.



Note Tasks marked with an asterisk (*) are mandatory.

Table 2: PacketCable Basic Workflow

	Task	Refer to...
Step 1	Configure the DPE	
	a. Configure a KDC service key.*	The service packetcable 1..1 registration kdc-service-key command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
	b. Enable PacketCable.*	The service packetcable 1..1 enable command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
Step 2	Configure DHCP	
	a. Configure dynamic DNS for the MTA scopes.	Cisco Prime Network Registrar 8.1 User Guide
	b. Configure client classes/scope-selection tags that match those added for provisioned PacketCable MTA DHCP criteria.*	Cisco Prime Network Registrar 8.1 User Guide
Step 3	Configure DNS	
	Configure dynamic DNS for each DHCP server.	Cisco Prime Network Registrar 8.1 User Guide
Step 4	Configure a Class of Service, which must contain the following properties:	
	<p>a. <i>/pktcbl/prov/flow/mode</i></p> <p>This property commands the specific flow that an MTA uses. Set this property to either:</p> <ul style="list-style-type: none"> • BASIC.1—Executes the BASIC.1 flow. • BASIC.2—Executes the BASIC.2 flow. <p>Note You can configure this property anywhere on the device-property hierarchy.</p>	Configuring Class of Service

	Task	Refer to...
	<p>b. <code>/cos/packetCableMTA/file</code></p> <p>This property contains the name of the configuration file that is to be presented to the MTA. The configuration file is stored as a file in Prime Cable Provisioning.</p> <p>The configuration file presented to a Basic MTA must contain the Basic integrity hash. If you are using a dynamic configuration template, the hash is inserted transparently during template processing. You can use the dynamic template for provisioning in both Secure and Basic modes.</p> <p>However, if the file is a Secure static configuration file, you must convert this file to a Basic static configuration file because Secure and Basic static configuration files are not interoperable. For details on how to perform this conversion, see Activating PacketCable Basic Flow.</p>	Configuring Class of Service

Configuring PacketCable Basic

Prime Cable Provisioning also supports PacketCable Basic, which offers a simpler, DOCSIS-like, non-secure provisioning flow. The following table describes the BASIC.1 flow using the provisioning workflow in [Figure 1: Embedded-MTA Secure Power-On Provisioning Flow, on page 12](#).

Table 3: PacketCable Basic eMTA Provisioning

Step	Workflow	Description
MTA-1	DHCP Broadcast Discover	Executes as for the Secure flow.
MTA-2	DHCP Offer	If the provisioning system is configured to provision the MTA in BASIC.1 mode, the provisioning system returns a DHCP Offer containing Option 122 suboption 6, which contains the special reserved realm name "BASIC.1". This reserved realm name commands the MTA to use the BASIC.1 provisioning flow. This Offer also contains the provisioning system IP address in Option 122.3, and the file and siaddr fields are populated with the configuration file location of the MTA.
MTA-3	DHCP Request	The remainder of the MTA DHCP exchange is executed (Request and Ack exchanged).
MTA-4	DHCP Ack	

Step	Workflow	Description
MTA-22	Telephony Config File Request	The MTA skips directly to step MTA-22. Using the file and siaddr information, the MTA copies its configuration file from the provisioning system via TFTP. Note that Prime Cable Provisioning integrates the TFTP server into the DPE component. Note No authentication of MTA/provisioning server or encryption occurs.
MTA-23	Telephony Config File	

The BASIC.2 flow is identical to BASIC.1, with the following exceptions:

- “BASIC.2” is populated into the MTA’s DHCP Option 122 suboption 6.
- The MTA issues a provisioning status SNMPv2c INFORM at the very end of the flow, MTA-25 (DHCP Option 122 suboption 3 specifies the Inform target).

The PacketCable Basic flow is similar to the DOCSIS flow with the following differences:

- There is no ToD exchange between MTA and the provisioning system.
- The MTA configuration file contains an integrity hash. Specifically, the SHA1 hash of the entire content of the configuration file is populated into a pkcMtaDevConfigFileHash SNMP VarBind and placed within a TLV 11 just before the end of file TLV.
- BASIC.2 flow issues a provisioning status SNMPv2c Inform after the MTA receives and processes its configuration file. This Inform notifies Prime Cable Provisioning if MTA provisioning completed successfully. If there is a problem, an error is generated and an event is sent from the DPE to the RDU, then on to a Prime Cable Provisioning client. This Inform is useful while debugging configuration file issues.

For additional information about the DOCSIS flow, see [Configuring DOCSIS, on page 1](#).


Note

Before using the PacketCable Basic provisioning flow, ensure that you are using a PacketCable Basic-capable eMTA. The eMTA must report that it is Basic-capable with its DHCP Discover Option 60, TLV 5.18 (supported flows).

PacketCable TLV 38 and MIB Support

Prime Cable Provisioning supports the complete set of PacketCable 1.5 MIBs.

Prime Cable Provisioning supports TLV 38 in PacketCable configuration templates. This TLV lets you configure multiple SNMP notification targets. Configuration of this TLV means that all notifications are also issued to the targets configured through TLV 38.

SNMP v2C Notifications

Prime Cable Provisioning supports both SNMP v2C TRAP and INFORM notifications from the PacketCable MTA.

PacketCable Secure

Prime Cable Provisioning supports two variants of PacketCable Secure:

- North American PacketCable
- European PacketCable

Euro-PacketCable services are the European equivalent of the North American PacketCable standard. The only significant difference between the two is that Euro PacketCable uses different MIBs. For details, see [Euro PacketCable](#), on page 22.

You perform the PacketCable-related tasks described in this section only after configuring the components as explained in [Configuring Prime Cable Provisioning Components](#).



Note

For PacketCable-compliant operations, the maximum allowable clock skew between the MTA, KDC, and DPE is 300 seconds (5 minutes). This value is the default setting.

The following table identifies the workflow to follow when configuring Prime Cable Provisioning to support PacketCable Secure.



Note

Tasks marked with an asterisk (*) are mandatory.

Table 4: PacketCable Secure Workflow

	Task	Refer to...
Step 1	Configure the RDU	
	a. Enable the autogeneration of Multimedia Terminal Adapter (MTA) FQDNs.	Automatic FQDN Generation
	b. Configure all provisioned DHCP Criteria.	Configuring DHCP Criteria
	c. Configure all provisioned Class of Service.	Configuring Class of Service
	d. Configure an SNMPv3 cloning key.*	Configuring SNMPv3 Cloning on RDU and DPE for Secure Communication with PacketCable MTAs , on page 21
	e. If you are using Euro PacketCable, configure the RDU to use Euro-PacketCable MIBs.	Euro PacketCable , on page 22
Step 2	Configure the DPE	

	Task	Refer to...
	a. Configure a KDC service key.*	The service packetcable 1..1 registration kdc-service-key command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
	b. Configure a privacy policy.*	The service packetcable 1..1 registration policy-privacy command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
	c. Configure an SNMPv3 cloning key.*	The service packetcable 1..1 snmp key-material command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
	d. Enable PacketCable.*	The service packetcable 1..1 enable command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
	e. Optionally, configure MTA file encryption.	The service packetcable 1..1 registration encryption enable command described in the Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide
Step 3	Configure the KDC	
	a. Obtain a KDC license from your Cisco representative.	KDC Certificate
	b. Configure a certificate chain using the PKCert.sh tool. For Euro PacketCable, use the -e option.	Using PKCert.sh
	c. Configure a service key pair for each DPE's provisioning FQDN.	Using KeyGen Tool
	d. Configure service keys for the ticket-granting-ticket (TGT).	Using KeyGen Tool
	e. Configure NTP Synchronization.	Solaris and Linux documentation for information on configuring NTP
Step 4	Configure DHCP	
	a. Configure all necessary PacketCable properties.	Using changeNRProperties.sh
	b. Configure dynamic DNS for the MTA scopes.	http://www.cisco.com/en/US/products/ps11808/products_user_guide_list.html

	Task	Refer to...
	c. Configure client classes/scope-selection tags to match those added for provisioned PacketCable MTA DHCP criteria.*	http://www.cisco.com/en/US/products/ps11808/products_user_guide_list.html
Step 5	Configure DNS	
	a. Configure dynamic DNS for each DHCP server.	http://www.cisco.com/en/US/products/ps11808/products_user_guide_list.html
	b. Configure a zone for the KDC realm.	http://www.cisco.com/en/US/products/ps11808/products_user_guide_list.html

Configuring PacketCable Secure

This section deals exclusively with Secure PacketCable voice provisioning. PacketCable Secure is designed to minimize the possibility of theft of telephony service, malicious disruption of service, and so on. PacketCable Secure depends on the Kerberos infrastructure to mutually authenticate the MTA and the provisioning system; in Prime Cable Provisioning, the Key Distribution Center (KDC) functions as the Kerberos server. SNMPv3 is also used to secure the conversation between the MTA and the provisioning system.



Note

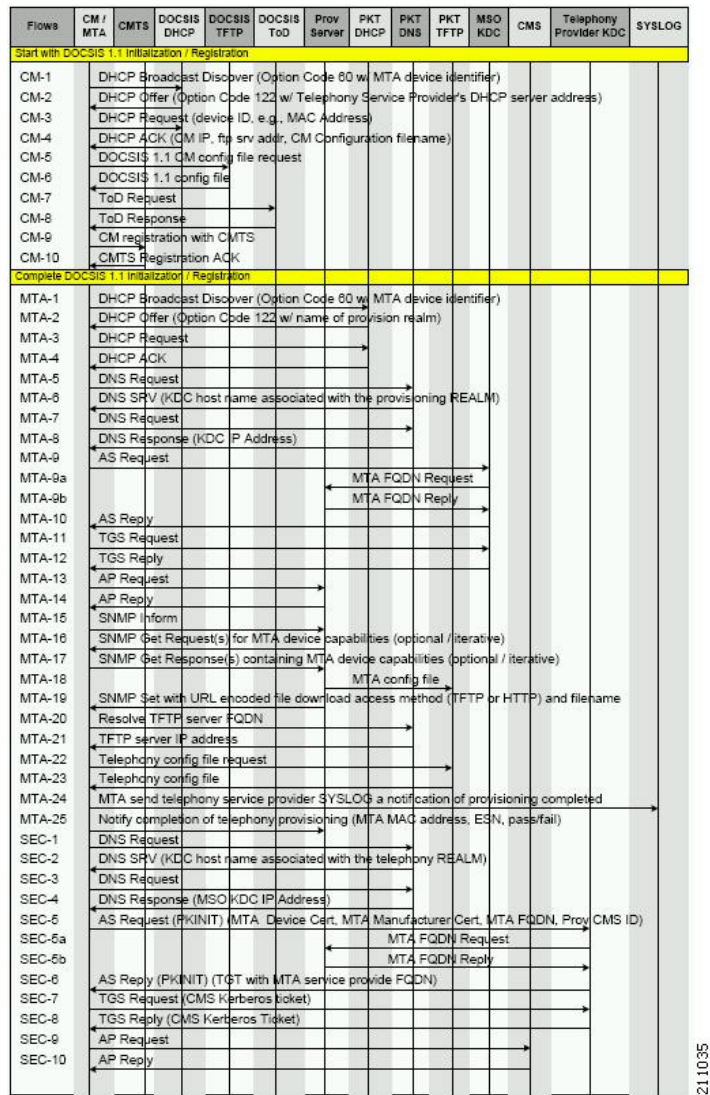
PacketCable secure provisioning is not supported for IPv6 devices.

Prime Cable Provisioning PacketCable Secure Provisioning Flow

All PacketCable provisioning flows are defined as a sequence of steps.

The following figure illustrates the Secure provisioning flow for PacketCable eMTAs.

Figure 1: Embedded-MTA Secure Power-On Provisioning Flow



Note

It is strongly recommended that you use a protocol analyzer (protocol sniffer) with the ability to capture data packets to understand exactly which step is failing.

In addition, the content of the KDC log file is critical to understanding the root cause of any KDC failure.

When diagnosing problems in provisioning an embedded Multimedia Terminal Adapters (eMTA), the flow description in the following table helps identify which step in the PacketCable provisioning flow is failing.

Table 5: PacketCable Secure eMTA Provisioning

Step	Workflow	Description
CM-1	DHCP Broadcast Discover	This is similar to the DOCSIS cable modem (CM) boot flow for DHCPv4 or DHCPv6 with DHCP options added to provide the MTA with a list of PacketCable DHCP servers from which the MTA is allowed to accept DHCP offers.
CM-2	DHCP Offer	
CM-3	DHCP Request	
CM-4	DHCP Ack	
CM-5	DOCSIS 1.1 CM Config File Request	
CM-6	DOCSIS 1.1 Config File	
CM-7	ToD Request	
CM-8	ToD Response	
CM-9	CM Registration with CMTS (cable modem termination system)	
CM-10	CMTS Registration Ack	

Step	Workflow	Description
MTA-1	DHCP Broadcast Discover	<p>Using DHCP, the MTA announces itself as a PacketCable MTA and provides information on the capabilities and provisioning flows it supports (Secure, Basic, and so on.). The MTA also obtains addressing information and DHCP Option 122. DHCP Option 122 contains the PacketCable provisioning server address and the security realm name. This information is used to allow the MTA to contact the KDC and provisioning server.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Check the DHCP relay agent on the CMTS for the correct configuration; ensure that your CMTS points to the correct DHCP server. • Verify that you have the correct routing between the MTA, CMTS, DHCP server, and the DPE. • Verify that secondary subnets are configured correctly on the CMTS. • Check the Prime Network Registrar DHCP configuration. Verify if the scopes are configured, if IP addresses are available, and if all secondary subnets are configured. • Check the Prime Cable Provisioning configuration. Check the <i>cnr_ep.properties</i> file and ensure that the required PacketCable Network Registrar extension properties are configured. For more information, see Mapping PacketCable DHCP Options to Prime Cable Provisioning Properties. <p>If a packet trace reveals that the MTA is cycling between steps MTA-1 and MTA-2, there could be a problem with the configuration of DHCP Option 122 (realm name or provisioning server FQDN suboptions), DHCP Option 12 (hostname), or DHCP Option 15 (domain name).</p>
MTA-2	DHCP Offer	
MTA-3	DHCP Request	
MTA-4	DHCP Ack	
MTA-5	DNS Request	<p>MTA uses the security realm name (delivered within DHCP Option 122) to perform a DNS SRV lookup on the KDC service and then resolves the KDC IP address.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Use a packet sniffer to watch for misdirected or malformed DNS packets sent to the Network Registrar DNS. • Set the Network Registrar DNS log level to detailed packet tracing and verify what arrives there. • Check the DNS configuration—The DNS server specified in <i>cnr_ep.properties</i> must contain the realm zone, the SRV record, and the DNS 'A' record for the KDC.
MTA-6	DNS Srv	
MTA-7	DNS Request	
MTA-8	DNS Response	

Step	Workflow	Description
MTA-9	AS Request	<p>The AS-REQ request message is used by the KDC to authenticate the MTA.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Check the KDC log file to determine if the AS-REQ arrives and to observe any errors or warnings. • Check that the KDC is configured with the correct MTA_Root certificate. The Manufacturer and Device certificates sent by the MTA within the AS-REQ message must chain with the MTA_Root certificate installed at the KDC.
MTA-9a	MTA FQDN Request	<p>The KDC extracts the MTA MAC address from the MTA certificate and sends it to the provisioning server for validation. If the provisioning server has the FQDN for that MAC address, it is returned to the KDC. The KDC then compares the FQDN received from the MTA to the FQDN received in the FQDN-REP reply message.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Use a packet sniffer to watch for misdirected or malformed DNS packets. The MTA passes the provisioning server FQDN (which the MTA received in DHCP Option 122) within the AS-REP message to the KDC. The KDC then uses this FQDN to resolve the IP address of the provisioning server. • Check the filenames and content of the KDC key file; the KDC service key in the DPE must match the service key at the KDC. The names of the service key files at the KDC are critical.
MTA-9b	MTA FQDN Reply	
MTA-10	AS Reply (AS-REP)	<p>The KDC grants a provisioning service ticket to the MTA and also sends the Service Provider, Local System Provider (optional), and KDC certificate to the MTA. The MTA then verifies if the certificates sent by the KDC chain to the Service Provider Root certificate stored in the MTA. If these certificates do not chain, the MTA loops back to step MTA-1 of the provisioning flow. See Using PKCert.sh, for additional information on the <i>KDC.cer</i> file.</p> <p>A key troubleshooting hint: Verify if the KDC log files show that the AS-REP message was sent to the device. If a packet trace reveals the MTA is cycling between steps MTA-1 and MTA-10, there is a problem with the service provider certificate chain.</p>
MTA-11	TGS Request	<p>The MTA receives either a service ticket or a ticket-granting-ticket (TGT) following step MTA-10. If the MTA had obtained a TGT instead of a service ticket in step MTA-10, it contacts the ticket-granting-server (KDC) to obtain a service ticket.</p>
MTA-12	TGS Reply	<p>The KDC sends a service ticket in the TGS Reply to the MTA.</p>

Step	Workflow	Description
MTA-13	AP Request (AP-REQ)	The MTA presents the ticket (received at step MTA-10) to the provisioning server specified by DHCP Option 122.
MTA-14	AP Reply (AP-REP)	The provisioning server uses the KDC shared secret to decrypt the AP-REQ, validates the provisioning server ticket presented by the MTA, and sends AP-REP with SNMPv3 keys. SNMPv3 is now authenticated and (optionally) encrypted.
MTA-15	SNMP Inform	The MTA signals to the provisioning server that it is ready to receive provisioning information.
MTA-16	SNMP Get Request	SNMPv3—If the provisioning server (DPE) requires additional device capabilities, it sends the MTA one or more SNMPv3 Get requests to obtain the required information on MTA capability. The provisioning server (DPE) may use a GetBulk request to request a bulk of information in a single message.
MTA-17	SNMP Get Response	SNMPv3—The MTA sends to the provisioning server (DPE) a response for each GetRequest that contains information on MTA capabilities requested in step MTA-16.
MTA-18	MTA Config file	Using information made available in steps MTA-16 and MTA-17, the provisioning server (DPE) determines the contents of the MTA configuration data file.
MTA-19	SNMP Set	SNMPv3—The provisioning server performs an SNMPv3 Set to the MTA containing the URL for the MTA configuration file, encryption key for the file, and the file hash value.
MTA-20	Resolve TFTP Server FQDN	DNS Request—If the URL-encoded access method contains an FQDN instead of an IPv4 address, the MTA uses the DNS server of the service provider network to resolve the FQDN into an IPv4 address of the TFTP server or the HTTP server.
MTA-21	TFTP Server IP Address	DNS Response—The DNS server returns the IPv4 IP address of the service provider network as requested in step MTA-20.
MTA-22	Telephony Config File Request	The MTA proceeds to download the VoIP configuration file from the specified TFTP server. Note that Prime Cable Provisioning integrates the TFTP server into the DPE component.
MTA-23	Telephony Config File	
MTA-24	MTA Send	The MTA optionally sends a syslog notification to the service provider that provisioning is complete.
MTA-25	Notify completion of telephony provisioning	The MTA signals to the provisioning server if the new configuration is acceptable.

Step	Workflow	Description
SEC-1 to SEC-10		These steps are the post-MTA provisioning security flow and are not applicable to provisioning of Prime Cable Provisioning. This flow involves getting Kerberos tickets associated with each CMS with which the MTA communicates. For details, see the PacketCable Security Specifications.

Configuring SRV Records in the Prime Network Registrar DNS Server

You must configure the Prime Network Registrar DNS server to operate with the KDC. To set up this configuration, see Prime Network Registrar documentation and these instructions.



Note

We recommend that you create a zone name that matches the desired realm name, and that the only DNS record in this special zone (other than the records required by the DNS server to maintain the zone) should be the SRV record for the realm. This example assumes that the desired Kerberos realm is `voice.example.com`, and that all other KDC, Network Registrar, and DPE configurations have been performed. The FQDN of the KDC is assumed to be `kdc.example.com`.

Step 1 Start the **nrcmd** command-line tool (which resides, by default, in the `/opt/nwreg2/local/usrbin` directory).

Step 2 Enter your username and password.

Step 3 To create a zone for the Kerberos realm, enter:

```
nrcmd> zone voice.example.com create primary address_of_nameserver hostmaster
```

where *address_of_nameserver* specifies the IP address of the name server.

Step 4 To add the SRV record to the new zone, enter:

```
nrcmd> zone voice.example.com. addRR _kerberos._udp. srv 0 0 88 KDC_FQDN
```

where *KDC_FQDN* specifies the FQDN of the KDC.

Step 5 To save and reload the DNS server, enter:

```
nrcmd> save
```

```
nrcmd> dns reload
```

Configuring DHCPv6 Server Selection

Prime Cable Provisioning supports sub-option 123 of option 125 specified in RFC 3925 and sub-option 2171 of option 17 specified in RFC 3315 for provisioning PacketCable 2.0 devices. To provide server identification in DHCPv6, Prime Cable Provisioning uses the CableLabs-specific DHCP Server Selection Identifier. The eCM is provided with a primary and secondary DHCP Server Selection Identifier via sub-options 1 and 2 within DHCPv4 option CL_V4OPTION_CCCV6 (123) or DHCPv6 option CL_OPTION_CCCV6 (2171).

The value set for the DHCP Server Selection Identifier defines whether the device can provision or not. By default this value is set to `ff:ff:ff:ff` when packet cable is disabled. You can configure this value at the time of installation of CPNR-EP component or [Using `changeNRProperties.sh`](#).

For example, if the eCM obtains a value ff:ff:ff:ff in sub-option 1 of CL_V4OPTION_CCCV6 or CL_OPTION_CCCV6, then the eUE is free to accept a valid DHCPv6 Advertise from any server, regardless of that server's DHCP Server Selection Identifier. Similarly, a value of 00:00:00:00 indicates that the eUE will not provision.

For more information about option 17.2171 and 125.123, see [Option 17.2171 or 125.123 and Prime Cable Provisioning Property Comparison](#).

The DHCP options for DSS_ID and IP Preference will be added to the response only when the Provisioning group capability **IPv6 - PacketCable 2.0** is enabled (ProvGroupCapabilitiesKeys. PACKET_CABLE_V6).

The below options will be ignored (or filtered) while generating the DHCP instructions whenever the PG capability **IPv6 - PacketCable 2.0** is disabled.

- CL_V4OPTION_CCCV6(123)
- CL_V4OPTION_IP_PREF(124)
- CL_OPTION_CCCV6(2171)
- CL_OPTION_IP_PREF(39)



Note

The above options can be added to the DHCP instructions whenever **IPv6 - PacketCable 2.0** capability is enabled. However the inclusion/ignoring of these options will be controlled by the following properties:

- 1 /pktcbl/ipPreference
- 2 /pktcbl/dssid/processing/enable

The property, /pktcbl/dssid/processing/enable (PacketCableDefaultKeys. PKTCBL_OPTION_DSS_ID_PROCESSING_ENABLE) is available to control the inclusion of DSS_ID options while generating the DHCP instructions. If this Boolean property is disabled then the below options will be ignored (or filtered) while generating the DHCP instruction.

- CL_V4OPTION_CCCV6 (123)
- CL_OPTION_CCCV6(2171)

In admin UI, DSS_ID processing options can be set at Device, Class of Service, DOCSIS Defaults, PacketCable Defaults and DHCP Criteria level in the RDU using the property /pktcbl/dssid/processing/enable. By default, the DSS_ID processing option will be disabled.

Configuring IP Preference Options

Prime Cable Provisioning now supports DHCP IP preference options CL_V4OPTION_IP_PREF (125.124) and CL_OPTION_IP_PREF (17.39). The IP preference option is requested by the DOCSIS modem (eCM:EDVA) when it is provisioned in a network. These options indicate whether the eUE must operate in single stack mode or dual stack mode for most operations (e.g., media, SIP signaling). RDU assigns IP Preference value to the PacketCable device based on its single stack or dual stack capability. These DHCP IP preference options indicate if IPv4 or IPv6 address must used for the eUE provisioning.

In the Admin UI, IP preference value can be set at Device, Class of Service, DOCSIS Defaults, and DHCP Criteria level in the RDU using the property /pktcbl/ipPreference. This can also be configured from RDU

API at any acceptable point in the property hierarchy such as, Device, Provisioning Group, Class of Service, DHCP Criteria, and Technology Defaults. By default, the IP preference value at the RDU is set to 0.

The following table describes all the IP preference values that can be set in the RDU and the corresponding Provisioning Flow.

Table 6: IP Preference Value Configurable in the RDU

IP Preference value	Description
0	interpreted as null. The eUE is provisioned based on the default Provisioning Flow IP mode (IPv4 or IPv6).
1	eUE acquires only IPv4 address and IPv4 address is used for all its operations including Provisioning Flows.
2	eUE acquires only IPv6 address and IPv6 address is used for all its operations including Provisioning Flows.
5	eUE acquires both IPv4 and IPv6 address for its operations but only IPv4 address is used for Provisioning Flows.
6	eUE acquires both IPv4 and IPv6 address for its operations but only IPv6 address is used for Provisioning Flows.

The following table describes all the IP preference values that is sent from the device and the interpretation of the RDU for the corresponding values.

Table 7: IP Preference Value from the Device

IP Preference value	Description
null	Indicates that the device is not dual stack.
7 (b'111)	Indicates that the device is capable of provisioning in dual stack mode.

If the device does not support dual stack mode and IP Preference values set on RDU is dual stack mode value (5 or 6) then the IP preference value in the response DHCP packet will be adjusted to devices' capability.

For example, if device does not send any value for IP preference and IP Preference value set in the RDU is 5 or 6, then IP preference value sent in DHCP ack or reply packet will be set to the corresponding single stack mode values i.e., 1 or 2 respectively.

If IP preference is not set at the RDU and the device has not sent a IP preference value, RDU will ignore the generation of IP preference DHCP options for the eUE.

The following table describes the IP preference value sent in DHCP ack or reply packet depending on the IP preference values sent from the device and values set in the RDU.

Table 8: IP Preference Decision Matrix

PacketCable Dual-stack Enabled	PacketCable Dual-stack Disabled
--------------------------------	---------------------------------

Device Signal for IP Preference	RDU Property Value	Decision		Device Signal for IP Preference	RDU Property Value	Decision	
		eCM's DHCP Discovery IPv4 Flow	eCM's DHCP Discovery IPv6 Flow			eCM's DHCP Discovery IPv4 Flow	eCM's DHCP Discovery IPv6 Flow
null	0	Ignore generating DHCP instructions for Options 125.124 and 17.39 for IPv4 and IPv6 modes respectively.		null	0	Ignore generating DHCP instructions for Options 125.124 and 17.39 for IPv4 and IPv6 modes respectively.	
null	1	1		null	1	1	
null	2	2		null	2	2	
null	5	5		null	5	1	
null	6	6		null	6	2	
7	1	1	1	7	1	1	1
7	2	2	2	7	2	2	2
7	5	5	5	7	5	1	1
7	6	6	6	7	6	2	2
7	0	5	6	7	0	1	2

Adding a Dial Plan for PacketCable 2.0 Groovy

A dial plan is provisioned on the UE to inform the UE about how dialed digits should be interpreted. A dial plan is an ordered set of regular expressions combined with some special tokens that represent actions to be carried out by the UE when a regular expression is matched.

The dial plan is organized into a list of rules. The UE must apply the dial plan rules sequentially and upon matching a pattern, including timers, the UE must perform the specified action or actions.

To create a dial plan you must be familiar with the notation and content of Augmented Backus-Naur Form (ABNF) defined in RFC 4234. Below is a sample dial plan that you can use as a reference.



Note

If any issues are encountered while determining the device capabilities, Prime Cable Provisioning defaults to the Secure mode. While adding a dial plan, you can either use a groovy script or a binary file but not a template.

A sample groovy file to create a dial plan

```
def dialPlan = '''
    TIMER S=4.000000
    TIMER Z=2.000000
```

```

domain = "@ims.packetcable.com"
dialString = ";user=dialstring"
dialPhone = ";user=phone"

homeEmergencyNumber = "911"
localEmergencyNumber = "911"

MAP MainTable =
"0S" : MAKE-CALL
"0#" : MAKE-CALL
"00" : MAKE-CALL
"(=Emergency)" : EMERGENCY-CALL("sip:" "911" =domain =dialPhone)
"(=N11)" : MAKE-CALL("sip:" #1v =domain =dialString)
"(=SpeedDial)" : MAKE-CALL("sip:" #1v =domain =dialString)
"(=PhoneNumber)" : MAKE-CALL("sip:" #1v =domain =dialPhone)
"(=ImmediateVSCs)" : RETURN
"(=DelayedVSCs)" : RETURN
"(x{1-20})S" : MAKE-CALL("sip:" #1 =domain =dialPhone)
"(x{1-20})#" : MAKE-CALL("sip:" #1 =domain =dialPhone)
'''
* PKTC-IETF-MTA-MIB pktcMtaDevEnabled (1.3.6.1.2.1.140.1.1.6.0)
*/
configFile.add(TLV_SNMP("1.3.6.1.2.1.140.1.1.6.0", "Integer", "1"))
/*
* Device Level Configuration (Secure flow only):
* Include required Secure-flow realm TLVs
*/
if (isSecureProvFlowMode)
{
    // PKTC-IETF-MTA-MIB pktcMtaDevRealmName.1 (1.3.6.1.2.1.140.1.3.6.1.2.1)
    configFile.add(
        TLV_SNMP("1.3.6.1.2.1.140.1.3.6.1.2.1", "STRING", realmName))
    // PKTC-IETF-MTA-MIB pktcMtaDevRealmOrgName.1 (1.3.6.1.2.1.140.1.3.6.1.5.1)
    configFile.add(
        TLV_SNMP("1.3.6.1.2.1.140.1.3.6.1.5.1", "STRING", realmOrgName))
}
configFile.add(option.createOptionValue(OptionSyntax.SNMP,"64",[".pktcEUESTDMValue.1","STRING",dialPlan]));

```

**Note**

The sample PacketCable 2.0 groovy script (example_edva.groovy) uses numeric OIDs for the PacketCable Secure-mode SNMP TLVs (pktcMtaDevRealmName, pktcMtaDevRealmOrgName).

If the TLV length of Option 64 exceeds 4500, you must update the property /default/asnParser/bufferLength=20000 in /opt/CSCOBac/api/conf/api.properties and in /opt/CSCOBac/rdu/conf/rdu.properties.

Configuring SNMPv3 Cloning on RDU and DPE for Secure Communication with PacketCable MTAs

Prime Cable Provisioning lets you enable an external network manager for SNMPv3 access to MTA devices. Additionally, the RDU is capable of performing SNMPv3 operations in a specific MTA.

To enable this capability, set the security key material at the DPEs and RDU. After the key material has been set, the Prime Cable Provisioning application programming interface (API) calls that are used to create cloned SNMPv3 entries are enabled.

**Note**

Enabling this capability impacts provisioning performance.

Creating the Key Material and Generating the Key

Creating the key material is a two-step process:

- 1 Run a script command on the RDU.
- 2 Run a CLI command on the DPE.



Note

This shared secret is not the same shared secret as the CMTS or the Prime Cable Provisioning shared secrets.

To create the key material:

Step 1

From the *BPR_HOME/rdu/bin* directory, run this script on the RDU:

```
# generateSharedSecret.sh password
```

where *password* is any password, from 6 to 20 characters, that you create. This password is then used to generate a 46-byte key. This key is stored in a file, called *keymaterial.txt*, that resides in the *BPR_HOME/rdu/conf* directory.

Step 2

Run the **service packetcable 1..1 snmp key-material** DPE CLI command, with the *password* used in Step 1 to generate that key, on all DPEs for which this voice technology is enabled. This command generates the same 46-byte key on the DPE and ensures that the RDU and DPEs are synchronized and can communicate with the MTA securely. For details about the commands, and the specific security privileges to run these commands, see the [Cisco Prime Cable Provisioning 5.3 DPE CLI Reference Guide](#).

Euro PacketCable

Euro-PacketCable services are essentially the European equivalent of North American PacketCable services with the following differences:

- Euro PacketCable uses different MIBs.
- Euro PacketCable uses a different set of device certificates (*MTA_Root.cer*) and service provider certificates (Service Provider Root).

For Euro-PacketCable certificates, the *kdc.ini* file must have the *euro-packetcable* property set to true. The KDC supports Euro-PacketCable (tComLabs) certificate chains. The following is a sample Euro PacketCable-enabled KDC configuration file.

```
[general]
interface address = 10.10.10.1
FQDN = servername.cisco.com
maximum log file size = 10000
n saved log files = 100
log debug level = 5 minimum
ps backoff = 150 maximum
ps backoff = 300
euro-packetcable = true
```

When using Euro PacketCable, ensure that the value of the PacketCable property */pktcbl/prov/locale* is set to EURO. The default is NA (for North America). You can specify the locale in the Configuration File utility. See [Using Configuration File Utility for Template](#), for more information.

Euro-PacketCable MIBs

Euro-PacketCable MIBs are essentially snapshots of draft-IETF MIBs. MTA configuration files consist of SNMP VarBinds that reference the MIBs. There are substantial differences between the North American PacketCable and Euro-PacketCable MIBs; therefore, the North American PacketCable and Euro-PacketCable configuration files are incompatible. During installation, sample files for North American PacketCable (*cw29_config.tmpl*) and Euro PacketCable (*ecw15_mta_config.tmpl*) are copied to the *BPR_HOME/rdu/samples* directory.

Prime Cable Provisioning ships with the following Euro-PacketCable MIBs:

- DOCS-IETF-BPI2-MIB
- INTEGRATED-SERVICES-MIB
- DIFFSERV-DSCP-TC
- DIFFSERV-MIB
- TCOMLABS-MIB
- PKTC-TCOMLABS-MTA-MIB
- PKTC-TCOMLABS-SIG-MIB

Configuring Euro-PacketCable MIBs

To configure Prime Cable Provisioning to use Euro-PacketCable MIBs, you must change the Prime Cable Provisioning RDU property that specifies the MIBs to be loaded. By default, this property contains the PacketCable MIBs.

You can change the property in one of the following ways:

- Modify *rdu.properties* and restart the RDU.
- On the Admin UI, navigate to **Configuration > Defaults > System Defaults** and replace the MIB list with the list shown below. You do not need to restart the RDU.
- Use the Prov API *changeSystemDefaults()* call. You do not need to restart the RDU.

The property name is */snmp/mibs/mibList* (properties file) or *SNMPPropertyKeys.MIB_LIST* (the Prov API constant name). The property value is a comma-separated value (CSV) consisting of the required MIB names, as shown:

```
/snmp/mibs/mibList=SNMPv2-SMI,SNMPv2-TC,INET-ADDRESS-MIB,CISCO-SMI,CISCO-TC,SNMPv2-MIB,
RFC1213-MIB,IANAifType
-MIB,IF-MIB,DOCS-IF-MIB,DOCS-IF-EXT-MIB,DOCS-BPI-MIB,CISCO-CABLE-SPECTRUM-
MIB,CISCO-DOCS-EXT-MIB,SNMP-FRAMEWORK-MIB,DOCS
-CABLE-DEVICE-MIB,DOCS-QOS-MIB,CISCO-CABLE-MODEM-
MIB,DOCS-IETF-BPI2-MIB,INTEGRATED-SERVICES-MIB,DIFFSERV-DSCP-TC,DIFFSERV
-MIB,TCOMLABS-MIB,PKTC- TCOMLABS-MTA-MIB,PKTC-TCOMLABS-SIG-MIB
```

Configuring DPoE

The DOCSIS Provisioning of Ethernet Passive Optical Network (DPoE) 1.0 is a standard for provisioning EPON access technology using the existing DOCSIS provisioning flow. DPoE network offers IP high speed data services equivalent to DOCSIS networks, where the DPoE network acts like a DOCSIS CMTS. The DPoE system and DPoE Optical Network Unit appear to act like a DOCSIS CM also known as virtual CM(vCM). Prime Cable Provisioning uses the existing DOCSIS device type for DPoE vCM devices. DPoE configuration files contain a mixture of DOCSIS and DPoE-specific TLVs.

From 5.3 release, PCP also supports DPoE 2.0. DPoE 2.0 specifications augment the DPoE 1.0 specifications to provide requirements for additional service capabilities and corresponding provisioning and network management capabilities. This simplifies the provisioning of complex network-wide services.

To identify a DPoE vCM, refer to the DHCP discover data captured under Request Dictionary displayed under Device Details page. Details similar to the following example show up and if the text in bold appear in page, then it is DPoE vCM.

Example:

```
v-i-vendor-opts = enterprise-id 4491, (oro 1 2)
chaddr = 00:00:00:00:0d:12
relay-agent-info = (circuit-id 1 80:01:03:ef), (remote-id 2 00:00:00:00:0d:12), (v-i-vendor
-opts 9 enterprise-id 4491, (cmts-capabilities 1 (docsis-version 1 03:00
), (dpoe-system-version 1 01:00), (dpoe-system-pbb 4 10248294639d, 1a9eb
ee4971b, 26d07cd85ab2, 33800cflabbb, 3b87c25dffbb, 47bd40a08f95, 4fc50b5
3a070, 5768bd554059, 591cf857aeal, 638c2d178f8f, 6d932a665ec9, 74efc6fc0
60b, 7a602d489587)))
relay-agent-circuit-id = 01:04:80:01:03:ef
client-id-created-from-mac-address = 0
dhcp-class-identifier = AIC Echo,docsis3.0:
hlen = 06
giaddr = 4.0.0.1
vendor-encapsulated-options = (device-serial-number 4 000000000d12), (hardware-version-number
5 v3.2.1
), (software-version-number 6 v1.0.2), (boot-rom-version 7 BOOT1.0), (ve
ndor-oui 8 000000), (vendor-name 10 XEROX CORPORATION), (dpoe-embedded-c
omponents-list 55 ECM)
dhcp-parameter-request-list = {1,3,6,7,12,15,51,54,4,2,67,66}
client-id = ff:00:00:00:00:03:00:01:00:00:00:00:0d:12
```

Sample DPoE Configuration file

The sample DPoE configuration files are available under the installed package at location:

- Static file: dpoe_vcm.cm -- /opt/CSCObac/rdu/samples/docsis
- Groovy file: example_dpoe_vcm.groovy -- /opt/CSCObac/rdu/samples/groovy
- Template file: dpoe_vcm.tmpl -- /opt/CSCObac/rdu/templates



Note

Prime Cable Provisioning supports provisioning of DPoE vCMs in IPv4 and IPv6 mode. Also, only downstream of computer devices from a DPoE vCMs is supported.

See [DPoE Option Support](#) for the DPoE TLVs.

Differences between DPoE and DOCSIS Provisioning

The provisioning of a DPoE vCM is nearly identical to the provisioning of DOCSIS CM. This allows existing DOCSIS-based back-office systems (such as provisioning servers) to support DPoE vCM provisioning with minimal changes. However, there are minor differences between DPoE vCM and DOCSIS CM provisioning:

- The DPoE specifications do not support IPv6 or PacketCable Voice services. Only IP (HSD) and MEF services are supported.
- The DPoE System (CMTS) supplies additional relay agent DHCP options. The DHCPv4 Relay Agent CMTS capabilities option contains additional sub-options (sub-option 2: DPoE System Version Number and sub-option 4: DPoE System DHCPv4 PBB service option).
- The DPoE vCM does not request ToD. The DPoE System (CMTS) supplies the time reference directly to the vCM.
- The DPoE vCM uses the same DHCPv4 option-60 value as DOCSIS 3.0 CM (i.e., docsis3.0). By itself, the DHCPv4 option-60 value is not sufficient to identify the device as a DPoE vCM.
- The DPoE vCM uses a new eSAFE DHCP option-43 sub-option 55 for specifying the list of eSAFE devices behind the eCM.
- The DPoE vCM supports new configuration file TLVs not supported by DOCSIS 3.0 MULPI (i.e., TLVs [22/23].14, [22/23].14.1, [22/23].14.2, [22/23].14.5, [22/23].14.6, [22/23].15, [22/23].15.1, [22/23].15.2).
- The DPoE vCM does not require or support all configuration file TLVs required by DOCSIS 3.0 MULPI. When the DPoE system encounters a TLV that is not supported, then the DPoE system ignores the TLV and allow the DPoE ONU to register normally.

DPoE Workflow

DPoE workflow is same as DOCSIS workflow. See [DOCSIS Workflow](#) for details.

Configuring CableHome

This section describes the activities that must be performed to ensure a satisfactory CableHome deployment. There are two versions of the CableHome technology: secure (SNMP) and non-secure (DHCP). This section deals exclusively with the non-secure version.

This section assumes that you are familiar with the contents of the CableHome Specification CH-SP-CH1.0-I05-030801.

CableHome Workflow

To successfully configure Prime Cable Provisioning for provisioning using the non-secure CableHome technology, you must perform the tasks described in [Configuring Prime Cable Provisioning Components](#), in addition to those described in this section.

The following table describes the tasks you must perform on Prime Cable Provisioning to support CableHome.

Table 9: CableHome Workflow

	Task	Refer to...
Step 1	Configure the RDU	
	a. Configure provisioned DHCP Criteria. Add all the DHCP Criteria that will be used by the non-secure CableHome devices that you will provision.	Configuring DHCP Criteria
	b. Configure provisioned Class of Service. Add the Class of Service that may be used by any provisioned non-secure CableHome device.	Configuring Class of Service
	c. Configure the promiscuous mode of operation.	System Defaults
Step 2	Configure the DPE	
Step 3	Configure Network Registrar	
	Configure the client classes/scope-selection tags to match those added for the provisioned non-secure CableHome DHCP Criteria.	Cisco Prime Network Registrar 8.1 User Guide

Configuring Prime Network Registrar

This section describes how to configure Prime Network Registrar, the cable modem configuration system (CMTS).

-
- Step 1** Create selection tags for provisioned and unprovisioned WAN-MAN and also for provisioned WAN-Data. Configure unprovisioned and provisioned client classes and scopes for cable modems, as specified in [Cisco Prime Network Registrar 8.1 User Guide](#).
- Step 2** Configure unprovisioned and provisioned client classes and scopes for WAN-MAN.
- Step 3** Configure provisioned client classes and scopes for WAN-Data.
- Step 4** Add routes to all the subnets.
-

Configuring RDU

To configure CableHome support on the RDU, perform these configurations:

Configuring CableHome WAN-MAN

- 1 Create a DHCP Criteria for the provisioned WAN-MAN. To do this, set the client class to a client-class name that is configured in the Network Registrar CableHome WAN-MAN.
- 2 Create a Class of Service for the provisioned WAN-MAN.
 - Set the `/cos/chWanMan/file` to a CableHome configuration file appropriate for the Class of Service.
 - Set the `/chWanMan/firewall/file` to the desired firewall configuration file.

Configuring CableHome WAN-Data

Configure these WAN-Data parameters whenever you want portal services to obtain the WAN-Data IP addresses:

- 1 Create DHCP Criteria for WAN-Data.
- 2 Create Class of Service for WAN-Data.

Configuring DPE

To configure the DPE to support the CableHome technology:

-
- | | |
|---------------|---|
| Step 1 | Open the CableHome device provisioning WAN-MAN config file and verify that DHCP Option 60 is set to either CableHome1.0 or CableHome1.1. Some manufacturers use a proprietary MIB object to instruct a device to behave as a pure cable modem, a non-CableHome router, or a CableHome router. The device appears as a Computer whenever the device DHCP packet does not contain CableHome1.0 or CableHome1.1 in the DHCP Option 60. |
| Step 2 | If you want the portal services to obtain IP addresses for WAN-Data: <ul style="list-style-type: none">• Ensure that the WAN-MAN configuration file contains TLV 28 that sets <code>cabhCdpWanDataIpAddrCount</code> to a value that is greater than 0.• In the cable modem configuration file, set the maximum number of devices to include the number of WAN-Data IP addresses. |
| Step 3 | To enable self-provisioning when the CableHome device boots: <ul style="list-style-type: none">• In the <code>unprov-wan-man.cfg</code> portal services configuration file, set the portal services in the passthrough mode.• In the cable modem configuration file, set the maximum number of devices to at least 2 to allow provisioning of the WAN-MAN and a computer. The computer can directly access sign-up web pages to be self-provisioned. |
-

