プライム ケーブル プロビジョニング コンポーネントの設定

この章は、各 プライム ケーブル プロビジョニング コンポーネントを プライム ケーブル プロビジョニング をサポートするテクノロジーに対して設定するため、従う必要があるワークフローを説明します。プライム ケーブル プロビジョニング を設定する前にこれらの設定を実行し、特定のテクノロジーをサポートする必要があります。

下で指定されている順番で、プライム ケーブル プロビジョニング コンポーネントを設定する必要があります。

- 地域配信ユニットの設定 （1 ページ）
- プロビジョニングの Web サービスの設定 （9 ページ）
- デバイス プロビジョニング エンジンの設定 （12 ページ）
- Cisco Prime Network Registrar の設定 （15 ページ）
- キー発行局 （22 ページ）

地域配信ユニットの設定

このセクションでは、地域配信ユニット（RDU）コンポーネントを設定する方法について説明します。その他のコンポーネントおよびテクノロジーを設定する前に、これらの設定タスクを実行する必要があります。

次の表では、RDU を設定する際に準拠するワークフローを確認できます。

表 *1 : RDU* 設定ワークフロー

	タスク	参照先
ステップ 1	プライム ケーブル プロビジョニング で使用するシステム syslog サービスを設定します。	アラートを受信するために Syslog ユーティリティを設定する

	タスク	参照先
ステップ 2	プライム ケーブル プロビジョニング Admin UI にアクセスします。	管理者ユーザ インターフェイス
ステップ 3	ログイン パスワードを変更します。	Admin UI へのアクセス
ステップ 4	ライセンス ファイルを追加します。	プライム ケーブル プロビジョニングのライセンス キー
ステップ 5	RDU SNMP エージェントを設定します。	Using snmpAgentCfgUtil.sh Tool
ステップ 6	デフォルトの重大度ログ レベル、通知レベルを設定します。	RDU ログ レベル ツールの使用
ステップ 7	IPv4 または IPv6 のプロビジョニング グループ機能を有効にします。	プロビジョニング グループのモニタリング

RDU 拡張機能の設定

カスタム拡張ポイントの作成が、プライム ケーブル プロビジョニング Admin UI で使用される場合に可能なプログラミング アクティビティであり、プライム ケーブル プロビジョニング 動作を併用したり、新しいデバイスのテクノロジーへサポートを追加できます。

拡張機能の管理を十分に理解する前に、プライム ケーブル プロビジョニング が必要とする RDU 拡張を知る必要があります。バッチに代わってデバイスを中断する場合、少なくとも 1 個の中断拡張が関連付けられたテクノロジーの中断拡張ポイントに接続されている必要があります。

次の表は、プライム ケーブル プロビジョニング が拡張機能を実行するために必要な RDU 拡張ポイントを一覧にしています。

表 *2 :* 必要な *RDU* 拡張ポイント

拡張ポイント	説明	用途	特定のテクノロジー？
一般的な設定生成	デバイスの設定を生成するために実行します。テクノロジー固有のサービス レベル選択拡張機能の後や、テクノロジー固有の設定生成拡張機能の前に、この拡張ポイントに接続されている拡張機能が実行されます。このリリースに組み込まれたデフォルトの拡張機能では、この拡張ポイントを使用しないでください。	任意	非対応
設定の生成	デバイスの設定を生成するために実行します。	必須	対応
デバイスの検出	デバイスの DHCP 検出要求パケットの情報に基づき、デバイステクノロジーを決定するために実行します。	必須	非対応
中断	デバイスの中断を実行します。	必須	対応
パブリッシュ	外部データストアにプロビジョニングデータを公開するために実行します。プライム ケーブルプロビジョニングに組み込まれているデフォルトの拡張機能には、パブリッシングプラグインを含めないでください。	任意	非対応

拡張ポイント	説明	用途	特定のテクノロジー？
サービス レベルの選択	サービス レベルを選択してデバイスに付与するために実行します。この拡張ポイントに接続されている拡張機能は、一般的な設定生成拡張機能やテクノロジー固有の設定生成拡張機能の前に実行されます。	オプション	○
認証	リモート認証サーバ経由でユーザーを認証するために実行します。拡張機能は、RDU のデフォルトページに記載されている認証モードに基づき拡張ポイントに接続されます。Radius 拡張機能は、プライム ケーブル プロビジョニングの認証拡張機能にデフォルトで組み込まれています。	必須	対応

拡張機能の管理には次が含まれています。

- 新規クラスの作成 （4 ページ）
- RDU カスタム拡張機能ポイントのインストール （6 ページ）
- RDU 拡張機能の表示 （6 ページ）
- IPDeviceKeys プロパティによって異なる RDU 拡張機能 （6 ページ）

（注） カンマ区切りリストで拡張ポイントを指定することで、複数の拡張ポイントを指定できます。

新規クラスの作成

この手順は、カスタム拡張作成プロセス全体をより分かりやすく説明するために含まれています。さまざまな種類の拡張機能を作成することができます。この手順の目的に従い、新しいパブリッシング拡張ポイントが使用されます。

新規クラスを書き込むには。

ステップ 1 カスタム パブリッシング拡張機能の Java ソース ファイルを作成し、コンパイルします。

ステップ 2 拡張クラスを含む JAR ファイルのマニフェスト ファイルを作成します。

（注） マニフェスト ファイル作成とコマンドライン JAR ツールの使用に関する詳細は、Java のマニュアルを参照してください。

次に例を示します。

```
Name: com/cisco/support/extensions/configgeneration
Specification-Title: "DOCSIS TOD synchronization"
Specification-Version: "1.0"
Specification-Vendor: "General Cable, Inc."
Implementation-Title: "Remove the time-servers DHCP option"
Implementation-Version: "1.0"
Implementation-Vendor: "Cisco Systems, Inc."
```

（注） Java JAR ファイルのマニフェストには、名前値ペアとして形式化されている属性を含み、パッケージ バージョン情報を提供する属性グループをサポートします。プライム ケーブル プロビジョニング はこの情報が含まれていない拡張 JAR ファイルを受け入れる際、カスタム RDU 拡張を追跡するファイルにバージョン情報を含むマニフェストを含めることをお勧めします。

[Servers（サーバ）] > [Regional Distribution Unit（地域配信ユニット）] > [View Regional Distribution Unit Details（地域配信ユニット詳細の表示）] ページ経由で、Admin UI からマニフェスト情報を表示できます。デバイス統計セクションの後にインストールされている拡張 JAR ファイルとロードの拡張クラス ファイルに関する詳細情報が表示されます。また、RDU ログからもマニフェスト情報を表示できます。

ステップ 3 カスタム拡張ポイントの JAR ファイルを作成します。

次に例を示します。

```
C:\>jar cm0vf manifest.txt removetimeservers.jar com
added manifest
adding: com/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/support/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/support/extensions/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/support/extensions/configgeneration/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/support/extensions/configgeneration/
RemoveTimeServersExtension.class(in = 4038) (out= 4038)(stored 0%)
C:\>
```

（注） JAR ファイルには任意の名前を付けることができます。名前は記述可能ですが、別の既存の JAR ファイル名とは重複しないでください。

デバイス検出プロセス

デバイスの検出プロセスには、3 つのフェーズがあります。

1. 初期設定フェーズでは、ロギング制御などデバイスや項目に関する基本情報を保持するさまざまな「house keeping」フィールドを初期化します。
2. 2 つ目のフェーズでは、検出情報の収集で構成されています。デバイス検出に関連する情報は、DHCPv4 および DHCPv6 構成から抽出します。クラス id、リレー エージェント回線 id、リモート id 等の DHCPv4 情報、ベンダー固有の情報、そしてベンダー クラスやベンダー選択などの DHCPv6 情報等が収集されます。
3. 最後のフェーズは、収集されたすべての情報を確認し、デバイス タイプとこのデバイスの前に別のデバイスがあるか判断します。決定可能な場合は、デバイス検出コンテキストで設定されます。

RDU カスタム拡張機能ポイントのインストール

JAR ファイルが作成された後、Admin UI を使用してインストールします。

ステップ 1 新しい JAR ファイルを追加するには、ファイルの追加 を参照してください。

（注） JAR ファイル タイプを選択します。[Browse] を使用して、新規クラスの作成 （4 ページ） で説明されている手順で作成された JAR ファイルを検索し、送信元ファイルとしてこのファイルを選択します。[Leaving the File Name] を空白にすると、送信元とファイルの両方に同じファイル名を割り当てます。ファイル名は、Admin UI で表示されます。

ステップ 2 [Submit] をクリックします。`

ステップ 3 RDU デフォルト ページに戻り、[Extension Point JAR File Search Order] フィールドに新しく追加された JAR ファイルが表示されているか確認します。

ステップ 4 [Publishing Extension Point] フィールドに拡張子クラス名を入力します。

（注） JAR ファイル内にクラス名が存在しなかった場合、RDU はエラーを返します。例えば、設定したクラスが置換 JAR ファイルになかった場合など、JAR ファイルを置換したときに主にこのエラーが発生します。

ステップ 5 **[Submit]** をクリックして、RDU データベースへの変更を確定します。

ステップ 6 RDU 拡張機能を表示して、適切な拡張機能がロードされたことを確認します。

RDU 拡張機能の表示

[View Regional Distribution Unit Details] ページから、すべての RDU 拡張機能の属性を直接表示できます。このページでは、インストールされている拡張 JAR ファイルと、ロードされた拡張クラス ファイルの詳細が表示されます。Monitoring RDUを参照してください。

IPDeviceKeys プロパティによって異なる RDU 拡張機能

RDU 組み込み拡張機能の動作は、プライム ケーブル プロビジョニング プロパティ階層で定義されている IPDeviceKeys プロパティによって異なります。たとえば、PacketCable BASIC vs. セ

キュア モード プロビジョニングは、サポートされているプロビジョニング フロー モードの DHCPv4 オプション 60 機能に基づいています。

表 *3 : IPDeviceKeys* プロパティによって異なる *RDU* 拡張機能

プロパティ	スコープ	RDU 組み込み拡張機能
EXTENDED_FILENAME_SCRIPT	DHCP	ConfigGeneration
DROP_IF_MAX_IA_ADDRESSES_EXCEEDED_ENABLE MAX_IA_ADDRESSES	DHCP	ConfigGeneration
MUST_BE_BEHIND_DEVICE MUST_BE_IN_PROV_GROUP MUST_BE_BEHIND_DEVICE_AUTO_ENABLE	DHCP	ServiceLevelSelection
USE_BOOT_FILE_OPTION_FOR_CONFIG_FILE USE_FILE_OPTION_FOR_CONFIG_FILE	DHCP	ConfigGeneration
USE_VALIDATE_CONTINUE_FOR_CABLELABS_SOFTWARE_VERSION_DHCPV4 USE_VALIDATE_CONTINUE_FOR_CABLELABS_SOFTWARE_VERSION_DHCPV6 USE_VALIDATE_CONTINUE_FOR_CLIENT_ID USE_VALIDATE_CONTINUE_FOR_CLIENT_ID_CREATED_FROM_MAC_ADDRESS USE_VALIDATE_CONTINUE_FOR_DHCP_CLASS_IDENTIFIER USE_VALIDATE_CONTINUE_FOR_DHCP_PARAMETER_REQUEST_LIST USE_VALIDATE_CONTINUE_FOR_VENDOR_CLASS USE_VALIDATE_PREFIX_FOR_DHCP_CLASS_IDENTIFIER USE_VALIDATE_CONTINUE_FOR_DHCP_CL_OPTION_IP_PREF_DHCPV6 USE_VALIDATE_CONTINUE_FOR_DHCP_CL_OPTION_IP_PREF_DHCPV4 USE_VALIDATE_CONTINUE_FOR_DHCP_CL_OPTION_ORO_DHCPV6 USE_VALIDATE_CONTINUE_FOR_DHCP_CL_OPTION_ORO_DHCPV4 USE_VALIDATE_CONTINUE_FOR_RELAY_AGENT_CIRCUIT_ID	DHCP	ConfigGeneration

プロパティ	スコープ	RDU 組み込み拡張機能
VALIDATE_CABLELABS_SOFTWARE_VERSION_DHCPV4 VALIDATE_CABLELABS_SOFTWARE_VERSION_DHCPV6 VALIDATE_CLIENT_ID VALIDATE_CLIENT_ID_CREATED_FROM_MAC_ADDRESS VALIDATE_DHCP_CLASS_IDENTIFIER VALIDATE_DHCP_PARAMTER_REQUEST_LIST VALIDATE_VENDOR_CLASS VALIDATE_DHCP_CL_OPTION_IP_PREF_DHCPV6 VALIDATE_DHCP_CL_OPTION_IP_PREF_DHCPV4 VALIDATE_DHCP_CL_OPTION_ORO_DHCPV6 VALIDATE_DHCP_CL_OPTION_ORO_DHCPV4 VALIDATE_RELAY_AGENT_CIRCUIT_ID	DHCP	ConfigGeneration

リモート SNMP リセットの設定

デフォルトのデバイス SNMP リセット（アクティベーション）は、中断の延長を使って RDU によって行われます。デバイス中断の実装は、RDU からデバイスに SNMP 設定メッセージを送信します。

プライム ケーブル プロビジョニング 6.1 は、デバイス SNMP リセット要求が RDU 以外に DPE から送信可能な場合、リモート SNMP リセットをサポートします。リセット操作中に、RDU は DPE にリセット要求を送信し、DPE は SNMP 設定をデバイスに送信します。

Admin UI または API を使用して、DPE 機能によりデバイス SNMP リセットを有効または無効にできます。この機能を有効化/無効化する PG 機能は次のとおりです。

- 機能名：**/provgroup/capability/dpe/remote/snmp/reset**
- API 定数：**ProvGroupCapabilitiesKeys.REMOTE_SNMP_RESET**

リセット操作から DPE の除外

SNMP リモート リセット機能が PG に対して有効な場合、デバイス リセット操作中に、RDU はリセット要求を PG の使用可能な DPE（MAC.DUID ベース アフィニティに基づく）のいずれかに送信します。ただし、ユーザーは DPE の除外リストを設定できるため、RDU はそれらの DPE にリセット要求を送信しません。

リモート リセットを送信中に PG レベルで特定の DPE を除外するプロパティは次のとおりです。

- 機能名：**/provgroup/capability/dpe/remote/snmp/reset/exclude/dpes/csv**
- API 定数：**ProvGroupCapabilitiesKeys.REMOTE_RESET_EXCLUDE_DPES_CSV**

プロパティの値は、コンマ区切り値（CSV）であり、リモート SNMP リセットで除外する必要がある DPE のホスト名で構成されます。

ケーブル モデム リセットのデフォルト SNMP バージョン

ケーブル モデム リセットは、デフォルトでは 1 の SNMP バージョンを使用して行われます。ユーザーはケーブル モデム リセットに SNMP v2c を使用するように設定できます。

（注） DOCSIS モデム リセットにデフォルトの SNMP バージョンを設定するには。

- API 定数：**ServerProperties.RDU_DOCSIS_RESET_SNMP_VERSION**
- プロパティ名：**/rdu/docsis/reset/snmp/version**

このプロパティは *rdu.properties* で設定でき、DOCSIS デバイスに SNMP デバイス リセット メッセージを送信中に使用可能な SNMP バージョンが含まれます。このプロパティのデフォルト値は 0（SNMP v1）であり、該当する値は（0 = SNMP v1; 1 = SNMP v2c）です。

プロビジョニングの Web サービスの設定

Provisioning Web Services（PWS）コンポーネントは、外部統合インターフェイスとして Web サービス インターフェイスに基づき SOAP/REST を公開します。Web サービスは RDU 上のレイヤであり、RDU またはリモート サーバとして同じサーバに展開可能ですが、リモート サーバに展開することをお勧めします。

各 PWS サービスについては、内部 API 要求が作成され、RDU に送信されます。サービスは 1 個以上の RDU と通信できます。

プロビジョニング サービスは、Web サービス記述言語（WSDL）v1.1 で説明されています。WSDL は Webサービス クライアントと RDU の間の契約であり、PWS 操作について説明します。SOAP メッセージをサポートする言語に、任意のクライアント言語のバインディングを生成するため PWS WSDL を使用することができます。

プロビジョニング サービスは、クライアントとのインタラクションのターゲットを識別するリソースのセットを使用して、RESTful Webサービスで説明されています。Webサービス クライアントと RDU 間の契約は URI で識別し、PWS RESTful 操作について説明します。

（注） PWS コンポーネントの共有秘密鍵はサポートされていません。

特定の PWS プロビジョニング機能は、ws cli.sh ツールを使用して実行することができます。ツールに関する詳細は、ws-cli.sh の使用 を参照してください。

次の表では、RDU を設定する際に準拠するワークフローを確認できます。

表 *4: PWS* 設定ワークフロー

	タスク	参照先
ステップ 1	RDU に接続する PWS を設定します。	PWS の RDU およびユーザーの詳細設定 （10 ページ）
ステップ 2	クライアントとの通信には、セッションを作成するか API を使用します。	デバイス プロビジョニング エンジンの設定 （12 ページ）

PWS の RDU およびユーザーの詳細設定

クライアントで要求されるすべてのプロビジョニング サービスを容易にするためには、PWS は RDU から情報を取得する必要があります。RDU と通信するために PWS は RDU にユーザー アカウントが必要であり、PWS のユーザー クレデンシャルと RDU の詳細を設定する必要があります。PWS をインストール中か、下に一覧になっている CLI コマンドを実行して、これらの詳細を PWS で設定可能です。

（注） RDU に追加されたユーザーは、ボックス Admin ロール外に必須です。

PWS で RDU の詳細を設定するには。

ステップ 1 CLI では、`-ardu <host> <port> <username> <Password>` コマンドを実行します。
次に例を示します。

```
./ws-cli.sh -ardu  test1-host  49187  admin  changeme
```

追加 RDU を追加するため同じコマンドを実行します。

ステップ 2 `-sap`、`--saveproperty` コマンドを実行して、ws.xml に入力したプロパティを保存します。
次に例を示します。

```
./ws-cli.sh -sap
```

ステップ 3 アカウントの作成後、変更が有効になっていることを確認するため PWS を再起動します。
次に例を示します。

```
/etc/init.d/bprAgent restart <pws|restpws>
```

Web サービスのプロビジョニングを使用して SOAP メッセージを投稿する手順

Web サービス クライアントは、次の形式の URL にアクセスして、PWS から WSDL（Web サービス契約）ファイルを取得できます。

http://<PWS-HOST>:<PWS-PORT>/cp-ws-prov/provService?wsdl

（注）
- PWS-HOST – PWS がインストールされているサーバのホスト名（または）IP アドレス
- PWS-PORT – PWS がアクセス可能な経由ポートセキュア モードの選択時、デフォルトの PWS HTTPS ポートは 9443 です。非セキュア モードの選択については、デフォルトの PWS HTTP ポートは 9100 です。ただし、ユーザーは PWS のインストール中にさまざまな（デフォルト以外の）HTTPS/HTTP ポートを設定できます。

SOAP メッセージを投稿するには、次の URL を使用する必要があり、WSDL が定義された PWS SOAP メッセージをロードする入力として提供されます。

http://<PWS-HOST>:<PWS-PORT>/cp-ws-prov/provService

（注）
- 「http」プロトコル ID は上記で参照した URL 形式で作成されたものとして使用されます。プロトコル識別子 https または http は、個別のセキュアまたは非セキュア通信に従って使用する必要があります。

Web サービスのプロビジョニングを通して RESTful メッセージを投稿する手順

Web サービス クライアントは、次の形式の URL にアクセスすることで、固有の URI にアクセスする必要があります。

http://<PWS-HOST>:<PWS-PORT>/cp-ws-rest-prov/<methodName >

（注）
- PWS-HOST – PWS がインストールされているサーバのホスト名（または）IP アドレス
- PWS-PORT – PWS がアクセス可能な経由ポートセキュア モードの選択時、デフォルトの PWS HTTPS ポートは 9790 です。非セキュア モードの選択については、デフォルトの PWS HTTP ポートは 9101 です。ただし、ユーザーは PWS のインストール中にさまざまな（デフォルト以外の）HTTPS/HTTP ポートを設定できます。

Restful メッセージを投稿するには、次の URL を使用する必要があり、要求本文の属性が定義された PWS Restful メッセージをロードする入力として提供されます。

http://<PWS-HOST>:<PWS-PORT>/cp-ws-rest-prov/<methodName >

（注）
- 「http」プロトコル識別子は、上記で参照されるURL形式で引用として使用されます。プロトコル識別子 https または http は、セキュアまたは非セキュア通信モードに従って使用される必要があります。

デバイス プロビジョニング エンジンの設定

表 1 : RDU 設定ワークフロー （1 ページ） に記載されている RDU の設定後のみ、このワークフローで説明されているタスクを実行します。下の表に示されるように、IPv4 および IPv6 をサポートする DPE を設定できます。

（注）アスタリスク（*）が付いている質問には必ず回答してください。

次の表では、IPv4 の DPE を設定するときに準拠するワークフローを特定します。

表 *5 : IPv4* の *DPE* 設定ワークフロー

	タスク	参照先
ステップ 1	プライム ケーブル プロビジョニングで使用するため、システム syslog サービスを設定します。	アラートを受信するために Syslog ユーティリティを設定する
ステップ 2	パスワードの変更	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **password** コマンド
ステップ 3	プロビジョニング インターフェイス* の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **interface ip** *ip_address* **プロビジョニング** コマンド
ステップ 4	プロビジョニング FQDN の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **interface ip** *ip_address* **provisioning fqdn** コマンド

	タスク	参照先
ステップ 5	Cisco Prime Network Registrar 拡張と通信するインターフェイスの設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **interface ip** *ip_address* **pg-communication** コマンド
ステップ 6	プライム ケーブル プロビジョニング共有秘密鍵*の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **dpe shared-secret** コマンド
ステップ 7	RDU * に接続するため DPE の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **dpe rdu-server** *{host | x.x.x.x} port secure* コマンド
ステップ 8	Network Time Protocol（NTP）の設定	設定情報に関する Linux マニュアル
ステップ 9	TFTP の有効化	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **service tftp** *1..1* **ipv4 enabled true** コマンド
ステップ 10	ToD の有効化	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **service tod** *1..1* **ipv4 enabled true** コマンド
ステップ 11	プライマリ プロビジョニング グループ*の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **dpe provisioning-group primary** コマンド
ステップ 12	DPE SNMP エージェントの設定	SNMP エージェント コマンド Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド
（注） DPE コマンドライン インターフェイスまたは **snmpAgentCfgUtil.sh** ツールのいずれかを使用して、SNMP エージェントを設定できます（Using snmpAgentCfgUtil.sh Tool を参照）。		

	タスク	参照先
ステップ 13	RDU に接続されていることを確認します	Admin UI を使用したサーバのモニタリング
ステップ 14	DPE のリロード	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **dpe reload** コマンド
ステップ 15	IPv4 のプロビジョニング グループ機能を有効にします	プロビジョニング グループのモニタリング

次の ***Table:15*** では、IPv6 の DPE を設定するときに準拠するワークフローを特定します。ここで説明されているタスクは、IPv6 のみに関連します。DPE の基本設定を実行するには、上記 ***Table 14: DPE Configuration Workflow for IPv4*** で説明されているタスクを完了し、さらにこの ***Table:15*** で説明されている手順を実行します。

表 *6 : IPv6* の *DPE* 設定ワークフロー

	タスク	参照先
ステップ 1	プロビジョニング インターフェイスの設定*。	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **interface ip** *ipv6_address* **provisioning** コマンド
ステップ 2	プロビジョニング FQDN の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **interface ip** *ipv6_address* **provisioning FQDNs** コマンド
ステップ 3	Cisco Prime Network Registrar 拡張と通信するインターフェイス（IPv6）の設定	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **interface ip** *ipv6_address* **pg-communication** コマンド
（注） 手順 3 はオプションです。この手順により、DPE との通信に IPv6 アドレスを使用する Network Registrar 拡張機能を有効にします。この手順は、DPE との通信に IPv4 アドレスを使用した Network Registrar 拡張機能は必要ありません。		

	タスク	参照先
ステップ 4	TFTP の有効化	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **service tftp** *1..1* **ipv6 enabled true** コマンド
ステップ 5	ToD の有効化	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **service tod** *1..1* **ipv6 enabled true** コマンド
ステップ 6	DPE のリロード	Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド で説明されている **dpe reload** コマンド
ステップ 7	IPv6 のプロビジョニング グループ機能を有効にします	プロビジョニング グループのモニタリング

Cisco Prime Network Registrar の設定

RDU と DPE を設定した後、このワークフローでのみ説明されているアクティビティを実行します。

注意 プライム ケーブル プロビジョニング DHCPオプション設定は、常に Prime Network Registrar 内の DHCP オプション値設定を置換します。

Network Registrar を設定するには。

- DHCPv4 は 表 7 : DHCPv4 の Network Registrar ワークフロー （16 ページ） を参照してください。
- DHCPv6 は 表 8 : DHCPv6 の Network Registrar ワークフロー （16 ページ） を参照してください。

（注） アスタリスク（*）が付いている質問には必ず回答してください。

次の表では、DHCPv4 の Network Registrar 設定時に従うワークフローを特定します。

表 *7 : DHCPv4* の *Network Registrar* ワークフロー

	タスク	参照先
ステップ 1	Network Registrar の拡張機能を検証します。	Cisco プライム ケーブル プロビジョニング 6.1.1 クイック スタート ガイド
ステップ 2	プライム ケーブル プロビジョニング で使用するシステム syslog サービスを設定します。	アラートを受信するために Syslog ユーティリティを設定する
ステップ 3	RDU で定義されているものと一致するクライアント クラス/選択タグを設定します。 *。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 4	ポリシーを設定します。*。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 5	範囲を設定します。*。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 6	Network Registrar データベースをバックアップします。	Cisco プライム ネットワーク レジスタ エンドユーザー ガイド
ステップ 7	適切な RDU に接続されていることを確認します。	Admin UI を使用したサーバのモニタリング
ステップ 8	DHCP サーバをリロードします。	Cisco プライム ネットワーク レジスタ エンドユーザー ガイド

次の表では、DHCPv6 の Network Registrar 設定時に従うワークフローを特定します。DOCSIS ケーブル モデム、コンピュータ、PacketCable MTA など、プロビジョニング済みおよび未プロビジョニング でバスの各カテゴリのタスク リストに従います。

表 *8 : DHCPv6* の *Network Registrar* ワークフロー

	タスク	参照先
ステップ 1	Network Registrar の拡張機能を検証します。	Cisco プライム ケーブル プロビジョニング 6.1.1 クイック スタート ガイド
ステップ 2	プライム ケーブル プロビジョニング で使用するシステム syslog サービスを設定します。	アラートを受信するために Syslog ユーティリティを設定する

	タスク	参照先
ステップ 3	RDU で定義されているものと一致するクライアント クラス/選択タグを設定します。 *。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 4	ポリシーを設定します。*。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 5	リンクを設定します。*。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 6	プレフィックスを設定します。各プレフィックスについては、適切なポリシー、リンク、および選択タグを設定していることを確認します。*。 （注） ケーブルモデムなどの一部の DHCP クライアントは、複数の IPv6 アドレスを含むオファーを拒否します。プレフィックスを定義する際に、クライアントに 1 つ以上の IPv6 アドレスを割り当てないように Network Registrar を設定します。Network Registrar が各プレフィックスから IP アドレスを 1 個選択し、クライアントに 2 個の IP アドレスを割り当てるため、2 個のプレフィックスに同じ選択タグを追加していないことを確認します。	Cisco Prime Network Registrar エンドユーザー ガイド
ステップ 7	Network Registrar データベースをバックアップします。	Cisco プライム ネットワーク レジスタ エンドユーザー ガイド

	タスク	参照先
ステップ 8	適切な RDU に接続されていることを確認します。	Admin UI を使用したサーバのモニタリング
ステップ 9	DHCP サーバをリロードします。	Cisco Prime Network Registrar エンドユーザー ガイド

Prime Network Registrarの拡張機能の設定

このセクションでは、Prime Network Registrar と通信するとき、プライム ケーブル プロビジョニング により使用される属性とオプションを説明します。

プライム ケーブル プロビジョニング そのデータベースにある設定に基づいて DHCP メッセージを操作するために、Prime Network Registrar でインストールされている DHCP 拡張機能を使用します。これらの拡張機能を使用して、プライム ケーブル プロビジョニング はDHCP 要求から情報を取得し、DHCP応答で値を設定します。この方法で、プロビジョニングするデバイスに対してカスタマイズされた設定を提供します。

この相互対話を容易するために、Prime Network Registrar は プライム ケーブル プロビジョニング 拡張機能にディクショナリのセットを公開します。プライム ケーブル プロビジョニング 拡張機能では、これらのディクショナリを使用して、Prime Network Registrar と相互対話します。

ディクショナリのタイプは 4 つあります。

- 環境ディクショナリ: DHCP サーバが拡張との通信に使用するディクショナリに含まれる属性を表します。
- 要求ディクショナリ: 要求パケットに対する DHCP オプションおよび属性を表します。
- 応答ディクショナリ: DHCP オプションおよび応答パケットの属性を表します。
- 通知ディクショナリ: プライム ケーブル プロビジョニング 拡張機能と RDU 間で通信する情報を表します。

ディクショナリは、プライム ケーブル プロビジョニング と Prime Network Registrarで設定されたさまざまな DHCP オプションと設定を表します。オプションは、DHCP メッセージのオプション フィールドに保存された DHCP 設定パラメータとその他び制御情報です。DHCP クライアントは、DHCP パケットで要求され、送信されるオプションを決めます。

属性は名前と値（name-value）のペアであり、次の場合があります。

- DHCPv4 オプション。例: **relay-agent-info**。
- DHCPv4 オプションから派生する情報のサブセット。例: **relay-agent-remote-id** は、DHCPv4 オプション 82 サブオプション 2 を表します。
- DHCPv4 オプションからのフィールド。例:「ファイル」は、DHCPv4 ヘッダー フィールドです。

属性には次のような設定も含まれています。

- Prime Network Registrar の動作を制御するもの。例:「drop」。パケットが廃棄されるようにことを示しています。
- 情報を提供するもの。

プライム ケーブル プロビジョニングは、Prime Network Registrar 9.x とともに、2 つの API バージョンをサポートし、プライム ケーブル プロビジョニング拡張機能のそれぞれは、DHCPv6 または DHCPv6 を有効にするために使用されます。

- DEX API バージョン 1: この API 属性によって、Prime Network Register 拡張機能で、属性経由で DHCPv4 パケットの詳細についてのクエリを行うことを許可します。
- DEX API バージョン 2: API は Prime Network Regstrar 拡張機能がDHCPv4 と DHCPv6 オプションとサブオプションに直接クエリすることを許可します。

  プライム ケーブル プロビジョニング 拡張機能が Prime Network Regstrar 拡張機能の API バージョンが DEX API バージョン 2 であることを検出する場合、DHCPv6 に対するサポートを有効にします。

DHCPv6 の検出されたデータを制御するプロパティ

プライム ケーブル プロビジョニング 拡張機能が DHCPv6 を検出するデータをコントロールするプロパティのセットは 3 つあります。

（注） Admin UI から、これらのプロパティの設定を **設定 > デフォルト > NR デフォルト** ページで表示することができます。

- バージョン プライム ケーブル プロビジョニング 以前の Prime Network Registrar の拡張機能の動作を制御するプロパティについては、表 9 : DHCPv4 Cisco Prime Network Registrar 拡張機能のプロパティ （20 ページ） を参照してください。
- プライム ケーブル プロビジョニング のDHCPv4 での Prime Network Registrar 拡張機能の動作を制御するプロパティは、表 9 : DHCPv4 Cisco Prime Network Registrar 拡張機能のプロパティ （20 ページ） を参照してください。
- クライアント (ケーブル モデム) とリレー エージェント (CMTS) に対する プライム ケーブル プロビジョニング の DHCPv6 に対する Prime Network Registrar の拡張機能の動作を制御するプロパティ。この違いは、DHCPv4 標準規格で、クライアントとリレー メッセージを 1 つのメッセージに組み合わせる一方で、DHCPv6 標準ではそれらを分割するために、発生します。表 10 : DHCPv6 Cisco Prime Network Registrar 拡張機能のプロパティ （20 ページ）を参照してください。

次の表に プライム ケーブル プロビジョニング バージョンの Prime Network Registrar の拡張機能の動作に影響するプロパティを示します。

表 *9 : DHCPv4 Cisco Prime Network Registrar* 拡張機能のプロパティ

プロパティ名	説明
/cnrExtension/attributesToReadFrom EnvironmentDictionary	Prime Network Registrar 環境ディクショナリから取得する必要がある属性のリストを特定します
	API 固定値 CNRExtensionSettingKeys.CNR_ATTRIBUTES_TO_READ_FROM_ENVIRONMENT_DICTIONARY
/cnrExtension/attributesRequiredInV4Request	Prime Network Registrar 要求ディクショナリには RDU に設定生成のための要求を送信する拡張機能を含まなければならない属性のリストを特定します。
	API 固定値 CNRExtensionSettingKeys.CNR_ATTRIBUTES_REQUIRED_IN_V4_REQUEST_DICTIONARY
/cnrExtension/attributesToPullFromV4RequestAsBytes	バイナリ形式で Prime Network Registrar 要求辞書から取得される属性のリストを特定します
	API 固定値 CNRExtensionSettingKeys.CNR_ATTRIBUTES_TO_READ_FROM_V4_REQUEST_DICTIONARY_AS_BYTES
/cnrExtension/attributesToPullFromV4RequestAsStrings	文字列形式で Prime Network Registrar 要求ディクショナリから取得される属性のリストを特定します
	API 固定値 CNRExtensionSettingKeys.CNR_ATTRIBUTES_TO_READ_FROM_V4_REQUEST_DICTIONARY_AS_STRINGS

次の表で プライム ケーブル プロビジョニング の DHCPv6 の Prime Network Registrar の拡張機能の動作を制御するプロパティを説明します。

表 *10 : DHCPv6 Cisco Prime Network Registrar* 拡張機能のプロパティ

プロパティ名	説明
クライアント メッセージ	

プロパティ名	説明
/cnrExtension/attributesRequiredInV6Request	Prime Network Registrar DHCPv6 要求ディクショナリには、RDU に設定生成の要求を送信するための拡張機能を含まなければならない属性のリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_ATTRIBUTES_REQUIRED_IN_V6_REQUEST_DICTIONARY`
/cnrExtension/attributesToPullFromV6RequestAsBytes	バイナリ形式で Prime Network Registrar DHCPv6 要求ディクショナリから取得される属性のリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_ATTRIBUTES_TO_READ_FROM_V6_REQUEST_DICTIONARY_AS_BYTES`
/cnrExtension/optionsRequiredInV6Request	Prime Network Registrar DHCPv6 要求ディクショナリが、RDU に設定の生成を要求を送信する拡張機能を含まなければならない DHCP オプションのリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_OPTIONS_REQUIRED_IN_V6_REQUEST_DICTIONARY`
/cnrExtension/optionsToPullFrom V6Request AsBytes	バイナリ形式で Prime Network Registrar DHCPv6 要求ディクショナリから取得される DHCP オプションのリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_OPTIONS_TO_READ_FROM_V6_REQUEST_DICTIONARY_AS_BYTES`
リレー メッセージ	
/cnrExtension/attributesRequiredInV6Relay	Prime Network Registrar DHCPv6 リレー転送要求ディクショナリには、RDU に設定生成の要求を送信するための拡張機能を含まなければならない属性のリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_ATTRIBUTES_REQUIRED_IN_V6_RELAY_DICTIONARY`

プロパティ名	説明
/cnrExtension/attributesToPullFromV6RelayAsBytes	バイナリ形式で Prime Network Registrar DHCPv6 リレー転送要求のリレーディクショナリから取得される属性のリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_ATTRIBUTES_TO_READ_FROM_V6_RELAY_DICTIONARY_AS_BYTES`
/cnrExtension/optionsRequiredInV6Relay	Prime Network Registrar DHCPv6 リレー転送要求ディクショナリが、RDU に設定の生成を要求を送信する拡張機能を含まなければならない DHCP オプションのリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_OPTIONS_REQUIRED_IN_V6_RELAY_DICTIONARY`
/cnrExtension/optionsToPullFromV6RelayAsBytes	バイナリ形式で Prime Network Registrar DHCPv6 リレー転送要求のリレーディクショナリから取得される DHCP オプションのリストを特定します
	API 固定値 `CNRExtensionSettingKeys.CNR_OPTIONS_TO_READ_FROM_V6_RELAY_DICTIONARY_AS_BYTES`

キー発行局

PacketCable Secure は Kerberos インフラストラクチャに依存して、MTA およびプロビジョニング システムを相互に認証します。プライム ケーブル プロビジョニング では、KDC は Kerberos サーバとして機能します。KDC コンポーネントの概要については Key Distribution Center（キー発行局） を参照してください。

デフォルト KDC プロパティ

KDC には、プライム ケーブル プロビジョニング のインストール中に *BPR_HOME/kdc/linux/kdc.ini* プロパティ ファイルに入力されるいくつかのデフォルト プロパティがあります。このファイルを編集して、運用要件の値を変更できます。

Note 運用の要件が指示されている場合、*Kdc.ini* ファイルの編集に注意してください。間違った値は KDC を動作しないようにレンダリングできます。変更する場合は、KDC を再起動します。

デフォルトのプロパティは次のとおりです。

- interface address：KDC により受信 Kerberos メッセージをモニタするローカル イーサネット インターフェイスの IP アドレスを指定します。

  次に例を示します。

  ```
  interface address = 10.10.10.1
  ```

- FQDN：KDC がインストールされる完全修飾ドメイン名（FQDN）を特定します。

  次に例を示します。

  ```
  FQDN = kdc.example.com
  ```

  Note インストール中に、KDC レルム名画面からインターフェイス アドレスおよびFQDN値を入力する必要があります。詳細については、Cisco プライム ケーブル プロビジョニング 6.1.1 クイック スタート ガイド を参照してください。

- maximum log file size：KDC によって生成されるログファイルが到達可能な最大サイズを指定します（キロバイト）。KDC は現在のファイルがこの最大サイズに達したときにのみ、新しいログファイルを作成します。

  次に例を示します。

  ```
  maximum log file size = 1000
  ```

- *n* saved log files：KDC が保存する古いログ ファイルの数を定義します。デフォルト値は 7 です。必要な数を指定できます。

  次に例を示します。

  ```
  n saved log files = 10
  ```

- log debug level：ログ ファイルのロギング レベルを指定します。

  ```
  log debug level = 5
  ```

次の表では、KDC ログ ファイルの使用可能なログ レベルについて説明します。

***Table 11: KDC* ロギング レベル**

ログ レベル	説明
0	エラー状態です。すべてのエラー メッセージとより深刻な性質のものを保存するロギング機能を設定します。
1	警告状態です。すべての警告メッセージとより深刻な性質のものを保存するロギング機能を設定します。
2	情報メッセージです。使用可能なすべてのロギング メッセージを保存するロギング機能を設定します。
{*3-7*}	デバッグ メッセージ。レベル3からレベル7まで、さまざまなレベルのすべてのデバッグ メッセージを保存するロギング機能を設定します。

- minimum (maximum) ps backoff : KDC が FQDN 要求に応答する プライム ケーブル プロビジョニング を待機する最小（最大）時間を秒単位で指定します。

  次に例を示します。

```
minimum ps backoff = 150
```

上記サンプルの値を使用して、サンプル INI ファイルは次の例に示すようなデータを含む可能性があります。

サンプル kdc.ini 設定ファイル

```
interface address =  10.10.10.1
FQDN = kdc.example.com
maximum log file size = 1000
n saved log files = 10
log debug level = 5
minimum ps backoff = 150
maximum ps backoff = 300
```

最少および最大チケット期間の時間を設定し、展開中に発生する可能性があるチケット要求の超過数を効果的にスムーズにできます。この設定は従来の稼働時間中に発生するほとんどの展開にお勧めで、常に過剰なローディングがパフォーマンスに悪影響を与える可能性があります。

Note チケット時間を短縮することで、MTA が KDC に対してより頻繁に認証するように強制します。テレフォニー エンドポイントの認証でより高度な制御が可能になる一方、KDC および向上したネットワーク トラフィックでより負荷の高いメッセージ ロードになる可能性があります。ほとんどの状況で、デフォルト設定が適切であり変更できません。

- チケットの最大期間：KDCによって生成されたチケットの最大時間を定義します。デフォルト単位は 1 時間です。**m** または **d** を追加して、個別に分または日に単位を変更できます。

  デフォルト値は168、または7日間です。この値はPacketCableセキュリティ仕様に準拠するために必要な時間の長さなため、この値は変更することをお勧めします。

  次に例を示します。

  ```
  maximum ticket duration = 168
  ```

- 最小チケット期間：KDC によって生成されたチケットの最小期間を定義します。デフォルト単位は 1 時間です。**m** または **d** を追加して、個別に分または日に単位を変更できます。

  デフォルト値は 144、または 6 日間です。この値を変更することは推奨しません。

  次に例を示します。

  ```
  minimum ticket duration = 144
  ```

KDC 証明書

KDC の認証に使用される証明書が プライム ケーブル プロビジョニング に同梱されていません。Cable Television Laboratories, Inc.（CableLabs）から必要な証明書を取得する必要があり、これらの証明書の内容が MTA にインストールされている証明書の内容と一致する必要があります。

Note 証明書は KDC 関数に必要です。

PKCert ツールを使用して、運用に必要な KDC の証明書をインストールおよび管理できます。PKCert ツールは、証明書ファイルとして CableLabs サービス プロバイダの証明書をインストールします。このツールを実行する方法については、PKCert.sh の使用 を参照してください。

PKCert ツールは、KDC コンポーネントをインストールしている場合にのみ使用できます。

KDC ライセンスのインストール

Cisco の担当者までから KDC ライセンスを取得し、適切なディレクトリにインストールします。

KDC ライセンス ファイルをインストールするには：

ステップ 1 Cisco 担当者でから、ライセンス ファイルを取得します。

ステップ 2 ルートとしてプライム ケーブル プロビジョニング ホストにログインします。

ステップ 3 *BPR_HOME/kdc* ディレクトリにライセンス ファイルをコピーします。

注意 ASCII ファイルとしてファイルをコピーしないように注意してください。ファイルには、ASCII 転送中に必要のない変更が行われやすいバイナリ データが含まれています。

転送プロセスがファイルに損害を与える可能性があるため、オペレーティング システム間で KDC ライセンス ファイルをコピーできません。

ステップ 4 KDC サーバを再起動し、変更を有効にするには、 **/etc/init.d** ディレクトリから *bprAgent restart kdc* コマンドを実行します。

追加のレルムの設定

プライム ケーブル プロビジョニング KDC は複数のレルム管理をサポートし、それに対して有効な PacketCable X.509 証明書および KDC 秘密キーの完全なセットが存在する必要があります。これらの証明書は、 *BPR_HOME/kdc/linux/packetcable/certificates* ディレクトリ内に存在する必要があります。

プライム ケーブル プロビジョニング は *BPR_HOME/kdc/linux/packetcable/certificates* ディレクトリのサブディレクトリをインストールして追加のレルムをサポートします。各サブディレクトリは特定のレルムの後に名前が付けられます。

次の表では、*BPR_HOME/kdc/linux/packetcable/certificates* ディレクトリに存在する対応するファイル名とともに、異なる証明書を一覧にしています。

表 *12 : PacketCable* 証明書

証明書	Certificate Filename
MTA ルート	*MTA_Root.cer*
サービス プロバイダ ルート	*CableLabs_Service_Provider_Root.cer*
サービス プロバイダ CA	*Service_Provider.cer*
ローカル システム オペレータ CA	*Local_System.cer*
KDC	*KDC*

プライマリ レルムは KDC コンポーネントのインストール時に設定されています。プライマリのレルムについては、KDC 証明書（*KDC.cer*）が *BPR_HOME/kdc/linux/packetcable/certificates* ディレクトリに存在します。その秘密キー（KDC_private_key.pkcs8）は *BPR_HOME/kdc/linux/* ディレクトリに存在します。

追加のレルムを設定するには、次で詳細を説明している手順に従います。

ステップ 1 KDC 証明書が含まれているディレクトリに移動します。

ステップ 2 KDC 証明書が保存されているディレクトリで、サブディレクトリを作成します。

（注） 特定のレルム名とサブディレクトリ名を一致させます。サブディレクトリに名前を付けるときは大文字のみを使用します。

ステップ 3 作成したサブディレクトリのレルムに KDC 証明書とレルムの秘密キーを配置します。

ステップ 4 新しいレルムが KDC 証明書と同じサービス プロバイダにチェーン接続されていない場合に、証明書ディレクトリとは異なるすべての高レベル追加証明書が含まれます。

（注） すべてのレルムが同じ証明書チェーンにルートされるため、KDC インストールでは任意の時点で 1 個のロケールのみサポートします（北米 PacketCable または欧州 PacketCable）。

次の表は、2 個のセカンダリ レルム（例：CISCO1.COM および CISCO2.COM）とともに、プライマリ レルム（例：CISCO.COM）のディレクトリ構造とファイルを説明します。構造では、高レベル証明書がプライマリ レルムおよびそのセカンダリ レルムと同様であると仮定します。

表 *13*：複数のレルムのディレクトリ構造

ディレクトリ	ディレクトリ内のファイル内容
BPR_HOME/kdc/linux/packetcable/certificates	プライマリ レルム CISCO.COM： • *MTA_Root.cer* • *CableLabs_Service_Provider_Root.cer* • *Service_Provider.cer* • *Local_System.cer* • *KDC* Directory /CISCO1.COM Directory /CISCO2.COM
*BPR_HOME/kdc/linux/packetcable/*証明書*/CISCO1*	セカンダリ レルム CISCO1.COM： • *KDC* • *KDC* 秘密キー

ディレクトリ	ディレクトリ内のファイル内容
BPR_HOME/kdc/linux/packetcable/certificates/CISCO2.COM	セカンダリ レルム CISCO2.COM： • *KDC* • *KDC* 秘密キー

複数のレルムの KDC 設定

このセクションでは、複数のレルムの KDC を設定するワークフローについて説明します。続行する前に、RDU、DPE、および Network Registrar 拡張機能のインストールを実行します。インストールの手順については、Cisco プライム ケーブル プロビジョニング 6.1.1 クイック スタート ガイド を参照してください。

次のワークフローでは、複数のレルムの KDC を設定する方法を説明するために、サンプルのレルムとディレクトリを使用します。ここで使用されるプライマリ レルムは CISCO.COM であり、そのセカンダリ レルムは CISCO1.COM および CISCO2.COM です。

次のワークフローで説明されているセットアップは、3 つの MTA をプロビジョニングします。Motorola SBV 5120 MTA、Linksys CM2P2 MTA、SA WebStar DPX 2203 MTA。各 MTA が 1 つのレルムにプロビジョニングされます。CISCO.COM レルムの Motorola、CISCO1.COM レルムに Linksys MTA、CISCO2.COM レルムの SA MTA。

（注） 次の手順に示す出力例は、デモ用にトリミングされています。

複数のレルムに KDC を設定するには。

ステップ 1 DPE で次の設定を検証します。

a) **Show run** コマンドを使用して、PacketCable サービスが有効であることを確認します。

PacketCable サービスを有効にするには、**service packetcable** *1..1* **enable** コマンドを使用します。

次に例を示します。

```
dpe# show run
  aaa authentication radius
  dpe port 49186
  dpe provisioning-group primary default
  service packetcable 1 enable
  snmp-server location equipmentrack5D
  snmp-server udp-port 8001
  tacacs-server retries 2
  tacacs-server timeout 5
```

コマンドの詳細については、Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド を参照してください。

b) **Show run** コマンドを使用して、KDC と DPE 間の通信に使用するセキュリティ設定を確認します。

セキュリティを生成し設定するには、**service packetcable** *1..1* **registration kdc-service-key** コマンドを使用します。

次に例を示します。

```
dpe# show run
  aaa authentication radius
  debug dpe events
  dpe port 49186
  service packetcable 1 enable
  service packetcable 1 registration kdc-service-key <value is set>
  snmp-server contact AceDuffy-ext1234
```

コマンドの詳細については、Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド を参照してください。

c) PacketCable SNMPv3 クローニングに DPE と RDU 間のセキュア通信を許可するセキュリティ キーが設定されていることを確認します。再度、**show run** コマンドを使用します。セキュリティを生成し設定するには、**service packetcable** *1..1* **snmp key-material** コマンドを使用します。

次に例を示します。

```
dpe# show run
  aaa authentication radius
  debug dpe events
  dpe port 49186
  service packetcable 1 enable
  service packetcable 1 registration kdc-service-key <value is set>
  service packetcable 1 snmp key-material <value is set>
```

コマンドの詳細と、これらのコマンドを実行する特定のセキュリティ権限については、Cisco プライム ケーブル プロビジョニング 6.1.2 リファレンス ガイド を参照してください。

（注） DPE で PacketCable を設定するときに、変更を有効にするため **dpe reload** コマンドを実行します。

ステップ 2 Network Registrar 拡張ポイント（cnr_ep.properties）の設定ファイルで、*/ccc/kerb/realm* パラメータがプライマリ レルムに設定されていることを確認します。この例では、CISCO.COM です。実行するには、**BPR_HOME/cnr_ep/conf** ディレクトリから *more cnr_ep.properties* コマンドを実行します。

次に例を示します。

```
/opt/CSCObac/cnr_ep/conf# more cnr_ep.properties
#DO NOT MODIFY THIS FILE.
#Tue Aug 13 23:24:00 PDT 2013
/ccc/tgt=01
/cccv6/dssid/primary=ff\:ff\:ff\:ff
/secure/keystore/file=/opt/CSCObac/lib/security/.keystore
/ccc/dhcp/primary=10.81.90.90
/secure/keystore/password=f2c2060fdbca0e60ae1864adb73155b9
/lib/cpcp/ssllib=/opt/nwreg2/local/lib/libssl.so.1.0.1
/rdu/fqdn=bactst-lnx-4
/server/rdu/secure/enabled=true
/rdu/port=49188
```

```
/cnr/sharedSecret=fgL7egT9zcYHs
/ccc/kerb/realm=CISCO.COM
/provgroup/capability/both/packetcable/ipv6=enabled
/provgroup/capability/both/packetcable/ipv4=enabled
/lib/cpcp/cryptolib=/opt/nwreg2/local/lib/libcrypto.so.1.0.1
/ccc/dns/primary=10.81.90.90
/cccv6/dssid/secondary=ff\:ff\:ff\:ff
/cnr/sharedSecret/digest=a3\:1f\:32\:6e\:57\:ed\:83\:b7\:68\:42\:f3\:31\:2b\:47\:d3\:36\:eb\:85\:93\:98
/cache/provGroupList=default
[root@bactst-lnx-7 ~]#
```

ステップ 3 静的ルートを適切に有効にして、CMTS の背後にあるデバイスと プライム ケーブル プロビジョニングを接続します。

ステップ 4 *Cnr_ep.properties* ファイルに記載されている DNS サーバの DNS レルムのゾーンを作成します。**[DNS] > [Forward Zones （前方ゾーン）] > [リスト/追加ゾーン（List/Add Zones）]** ページで Network Registrar Admin UI を使用して、ゾーンを追加できます。

（注） 追加するゾーンに KDC サーバの SRV レコードと DNS「A」レコードが含まれ、各ゾーンの（例：CISCO.COM、CISCO1.COM、CISCO2.COM）SRV レコードを 1 個の KDC にポイントします。

Admin UI からのゾーン設定の詳細については、『Cisco Prime Network Registrar End-User Guides』を参照してください。

ステップ 5 PKCert.sh ツールを使用して証明書を設定します。

a) *BPR_HOME/kdc/linux/packetcable/certificates*の下にセカンダリ レルムのディレクトリを作成します（例：CISCO1.COM および CISCO2.COM）

次に例を示します。

```
/opt/CSCObac/kdc/linux/packetcable/certificates# mkdir CISCO1.COM
/opt/CSCObac/kdc/linux/packetcable/certificates# mkdir CISCO2.COM
```

ディレクトリの作成に関する詳細は、Linux のマニュアルを参照してください。

b) 次の証明書をコピー可能なディレクトリを作成します。

- *CableLabs_Service_Provider_Root.cer*
- *Service_Provider.cer*
- *Local_System.cer*
- *MTA_Root.cer*
- *Local_System.der*

次に例を示します。

```
# cd /var
# mkdir certsInput
```

（注） */var* ディレクトリの下に作成した */certsInput* ディレクトリは例のみです。その他のディレクトリでディレクトリを作成するように選択できます。ディレクトリの作成に関する詳細は、特定のオペレーティング システムのマニュアルを参照してください。

c) 前の手順で記載した証明書を作成したディレクトリにコピーします。

d) 次の証明書を *BPR_HOME/kdc/linux/packetcable/certificates* ディレクトリにコピーします。

- *CableLabs_Service_Provider_Root.cer*
- *Service_Provider.cer*
- *Local_System.cer*
- *MTA_Root.cer*

ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

e) プライマリ レルムの KDC 証明書と関連付けられている秘密キーを作成します。

次に例を示します。

```
# ./opt/CSCObac/kdc/PKCert.sh -c "-s /var/certsInput -d /var/certsOutput
-k /var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer
-r CISCO.COM -n 100 -a bactest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: CISCO.COM
Serial Number: 100
DNS Name of KDC: bactest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObac/kdc/linux/
packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObac/
kdc/linux)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e. /opt/CSCObac/
kdc/linux)
```

ツールのに関する詳細については、PKCert.sh の使用 を参照してください。

f) KDC 証明書のディレクトリに KDC.cer ファイルをコピーします（*BPR_HOME/kdc/linux/packetcable/certificates*）。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

g) 秘密キー KDC_private_key.pkcs8 を KDC プラットフォーム ディレクトリ（*BPR_HOME/kdc/linux*）にコピーします。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

h) 秘密キー KDC_private_key を KDC プラットフォーム ディレクトリ（*BPR_HOME/kdc/linux*）にコピーします。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

i) セカンダリ レルムの KDC 証明書と関連付けられている秘密キーを作成します。この場合、CISCO1.COM です。

次に例を示します。

```
# ./opt/CSCObac/kdc/PKCert.sh -c "-s /var/certsInput -d /var/certsOutput
-k /var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer
-r CISCO1.COM -n 100 -a bactest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: CISCO.COM
Serial Number: 100
DNS Name of KDC: bactest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObac/kdc/linux/
packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObac/
kdc/linux)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e. /opt/CSCObac/
kdc/linux)
```

ツールのに関する詳細については、PKCert.sh の使用 を参照してください。

j) *KDC.cer* をセカンダリ レルム ディレクトリにコピーします。たとえば、*BPR_HOME/kdc/linux/packetcable/certificates* の下の */CISCO1.COM* ディレクトリです。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

k) 秘密キー KDC_private_key.pkcs8 をセカンダリ レルム ディレクトリにコピーします。たとえば、*BPR_HOME/kdc/linux/packetcable/certificates* の下の */CISCO1.COM* ディレクトリです。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

l) 秘密キー KDC_private_key_proprietary をセカンダリ レルム ディレクトリにコピーします。たとえば、*BPR_HOME/kdc/linux/packetcable/certificates* の下の */CISCO1.COM* ディレクトリです。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

m) セカンダリ CISCO2.COM レルムの KDC 証明書と関連付けられている秘密キーを作成します。

次に例を示します。

```
# ./opt/CSCObac/kdc/PKCert.sh -c "-s /var/certsInput -d /var/certsOutput
-k /var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer
-r CISCO2.COM -n 100 -a bactest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
```

```
Realm: CISCO.COM
Serial Number: 100
DNS Name of KDC: bactest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObac/kdc/linux/
packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObac/
kdc/linux)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e. /opt/CSCObac/
kdc/linux)
```

ツールのに関する詳細については、PKCert.sh の使用 を参照してください。

n) *KDC.cer* をセカンダリ レルム ディレクトリにコピーします。たとえば、*BPR_HOME/kdc/linux/packetcable/certificates* の下の */CISCO2.COM* ディレクトリです。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

o) 秘密キー KDC_private_key.pkcs8 をセカンダリ レルム ディレクトリにコピーします。たとえば、*BPR_HOME/kdc/linux/packetcable/certificates* の下の */CISCO2.COM* ディレクトリです。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

p) 秘密キー KDC_private_key_proprietary をセカンダリ レルム ディレクトリにコピーします。たとえば、*BPR_HOME/kdc/linux/packetcable/certificates* の下の */CISCO2.COM* ディレクトリです。ファイルのコピーに関する詳細は、**cp** コマンドで Linux マニュアルを参照してください。

ステップ 6 KeyGen ツールを使用して、PacketCable サービス キーを生成します。

（注） サービス キーを生成するために使用するパスワードが、**packetcable registration kdc service-key** コマンドを使用して DPE で設定したパスワードと一致していることを確認します。

次に例を示します。

```
# /opt/CSCObac/kdc/keygen bactest.cisco.com CISCO.COM changeme
# /opt/CSCObac/kdc/keygen bactest.cisco.com CISCO1.COM changeme
# /opt/CSCObac/kdc/keygen bactest.cisco.com CISCO2.COM changeme
```

詳細については、PKCert.sh の使用を参照してください。

ステップ 7 手順 6 で生成したサービス キーが *BPR_HOME/kdc/linux/keys directory* に存在することを確認します。

次に例を示します。

```
/opt/CSCObac/kdc/linux/keys# ls -l
total 18
-rw-r--r--   1 root     other 2 Nov  4 09:44 krbtgt,CISCO1.COM@CISCO1.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 krbtgt,CISCO2.COM@CISCO2.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 krbtgt,CISCO.COM@CISCO.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 mtafqdnmap,bactest.cisco.com@CISCO1.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 mtafqdnmap,bactest.cisco.com@CISCO2.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 mtafqdnmap,bactest.cisco.com@CISCO.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 mtaprovsrvr,bactest.cisco.com@CISCO1.COM
```

```
-rw-r--r--   1 root     other 2 Nov  4 09:44 mtaprovsrvr,bactest.cisco.com@CISCO2.COM
-rw-r--r--   1 root     other 2 Nov  4 09:44 mtaprovsrvr,bactest.cisco.com@CISCO.COM
```

詳細については、Linux マニュアルを参照してください。

ステップ 8 さまざまな証明書およびサービス キーが *BPR_HOME/kdc* ディレクトリが存在することを確認します。

次に例を示します。

```
/opt/CSCObac/kdc# ls
PKCert.sh  internal keygen lib pkcert.log  linux  bacckdc.license

/opt/CSCObac/kdc# cd /internal/bin
/internal/bin# ls
kdc runKDC.sh shutdownKDC.sh

# cd /opt/CSCObac/kdc/lib
# ls
libgcc_s.so.1      libstdc++.so.5     libstlport_gcc.so

# cd /opt/CSCObac/linux/logs
# ls
kdc.log    kdc.log.1

# cd /opt/CSCObac/linux
# ls
logs kdc.ini packetcable KDC_private_key_proprietary.

# cd keys
# ls
krbtgt,CISCO1.COM@CISCO1.COM
krbtgt,CISCO2.COM@CISCO2.COM
krbtgt,CISCO.COM@CISCO.COM
mtafqdnmap,bactest.cisco.com@CISCO1.COM
mtafqdnmap,bactest.cisco.com@CISCO2.COM
mtafqdnmap,bactest.cisco.com@CISCO.COM
mtaprovsrvr,bactest.cisco.com@CISCO1.COM
mtaprovsrvr,bactest.cisco.com@CISCO2.COM
mtaprovsrvr,bactest.cisco.com@CISCO.COM

# cd ./linux/packetcable/certificates
# ls
KDC.cer
Local_System.cer
CableLabs_Service_Provider_Root.cer  MTA_Root.cer
CISCO1.COM                           Service_Provider.cer
CISCO2.COM

# cd ./linux/packetcable/certificates/CISCO1.COM
# ls
KDC.cer
KDC_private_key_proprietary.

# cd ./linux/packetcable/certificates/CISCO2.COM:
# ls
KDC.cer
KDC_private_key_proprietary.
```

詳細については、Linux マニュアルを参照してください。

ステップ 9 KDC を再起動します。

次に例を示します。

```
# /etc/init.d/bprAgent restart kdc
```

詳細については、CLI からプライム ケーブル プロビジョニング プロセス ウォッチドッグを使用するを参照してください。

ステップ 10 複数のレルムのプライム ケーブル プロビジョニング Admin UI を設定します。

a) セカンダリ レルムの DHCP の条件を追加します。この例では、CISCO1.COM。

次に例を示します。

1. **[Configuration（設定）] > [DHCP Criteria（DHCP 条件）] > [Manage DHCP Criteria（DHCP 条件の管理）]** から、**[Add（追加）]** ボタンをクリックします。
2. [Add DHCP Criteria（DHCP 条件の追加）] ページが表示されます。
3. [DHCP Name（DHCP 名）] フィールドには、 **cisco1** と入力してください。
4. [Submit] をクリックします。`
5. [Manage DHCP Criteria（DHCP 条件の管理）] ページに戻り、cisco1 DHCP 条件をクリックします。[Modify DHCP Criteria（DHCP 条件の変更）] ページが表示されます。
6. プロパティ名の下の */ccc/kerb/realm* を選択し、CISCO1.COM をプロパティ値のフィールドに入力します。
7. **[Add（追加）]** および **[Submit（送信）]** をクリックします。

詳細については、DHCP 条件の設定を参照してください。

b) セカンダリ レルムの DHCP の条件を追加します。この例では、CISCO2.COM。

次に例を示します。

1. **[Configuration（設定）] > [DHCP Criteria（DHCP 条件）] > [Manage DHCP Criteria（DHCP 条件の管理）]** から、**[Add（追加）]** ボタンをクリックします。
2. [Add DHCP Criteria（DHCP 条件の追加）] ページが表示されます。
3. [DHCP Name（DHCP 名）] フィールドには、 **cisco2** と入力してください。
4. [Submit] をクリックします。`
5. [Manage DHCP Criteria（DHCP 条件の管理）] ページに戻り、cisco2 DHCP 条件をクリックします。[Modify DHCP Criteria（DHCP 条件の変更）] ページが表示されます。
6. プロパティ名の下の /ccc/kerb/realm を選択し、cisco2.COM をプロパティ値のフィールドに入力します。
7. **[Add（追加）]** および **[Submit（送信）]** をクリックします。

詳細については、DHCP 条件の設定を参照してください。

c) プロビジョニングされる各デバイスのプロビジョニング ケーブル プロビジョニングにファイルとしてテンプレートを追加します。この手順では、Motorola MTA です。

次に例を示します。

1. **[Configuration（設定）] > [Files（ファイル）]** を選択します。[Manage Files（ファイルの管理）] ページが表示されます。
2. **[Add（追加）]** をクリックすると、[Add Files（ファイルの追加）] ページが表示されます。
3. [File Type（ファイル タイプ）] ドロップダウンリストから CableLabs 設定テンプレートを選択します。
4. *Mot mta.tmp*l ファイルを追加します。このファイルは、Motorola MTA のプロビジョニングに使用されるテンプレートです。テンプレートの構文については、例 **Template Used to Provision a Motorola MTA** を参照してください。
5. [送信（Submit）] をクリックします。

詳細については、ファイルの管理を参照してください。

d) プロビジョニングされる各デバイスのプロビジョニング ケーブル プロビジョニングにファイルとしてテンプレートを追加します。この手順では、Linksys MTA です。

次に例を示します。

1. **[Configuration] > [Files]** を選択します。[Manage Files（ファイルの管理）] ページが表示されます。
2. **[Add（追加）]** をクリックすると、[Add Files（ファイルの追加）] ページが表示されます。
3. [File Type（ファイル タイプ）] ドロップダウンリストから CableLabs 設定テンプレートを選択します。
4. Linksys mta.tmpl ファイルを追加します。このファイルは、Linksys MTA のプロビジョニングに使用されるテンプレートです。テンプレートの構文については、例 **Template Used to Provision a Linksys MTA** を参照してください。
5. [送信（Submit）] をクリックします。

詳細については、ファイルの管理を参照してください。

e) プロビジョニングされる各デバイスのプロビジョニング ケーブル プロビジョニングにファイルとしてテンプレートを追加します。この手順では、 SA MTA です。

次に例を示します。

1. **[Configuration] > [Files]** を選択します。[Manage Files（ファイルの管理）] ページが表示されます。
2. **[Add（追加）]** をクリックすると、[Add Files（ファイルの追加）] ページが表示されます。
3. [File Type（ファイル タイプ）] ドロップダウンリストから CableLabs 設定テンプレートを選択します。

4. Sa mta.tmpl ファイルを追加します。このファイルは、SA MTA のプロビジョニングに使用されるテンプレートです。テンプレートの構文については、例 **Template Used to Provision a SA MTA** を参照してください。
5. [送信（Submit）] をクリックします。

詳細については、ファイルの管理を参照してください。

f) プライマリ レルムのサービス クラスを追加します。この例では、CISCO.COM。

次に例を示します。

1. **[Configuration（設定）] > [Class of Service（サービス クラス）]** の順に選択します。
2. [Add] をクリックします。[サービス クラスの追加（Add Class of Service）] ページが表示されます。
3. Mot mta として CISCO.COM レルムの新しいサービス クラスの名前を入力します。
4. PacketCableMTA としてサービス クラス タイプを選択します。
5. プロパティ名ドロップダウンリストから*/cos/packetCableMTA/file*を選択し、(プライマリ CISCO.COM レルムに Motorola MTA のプロビジョニングに使用) する mot mta.tmpl テンプレート ファイルを関連付けます。
6. **[Add（追加）]** および **[Submit（送信）]** をクリックします。

詳細については、サービス クラスの設定を参照してください。

g) セカンダリ レルムのサービス クラスを追加します。この例では、CISCO1.COM。

次に例を示します。

1. **[Configuration（設定）] > [Class of Service（サービス クラス）]** の順に選択します。
2. [Add] をクリックします。[サービス クラスの追加（Add Class of Service）] ページが表示されます。
3. linksys-mta として CISCO1.COM レルムの新しいサービス クラスの名前を入力します。
4. PacketCableMTA としてサービス クラス タイプを選択します。
5. プロパティ名ドロップダウンリストから*/cos/packetCableMTA/file*を選択し、(セカンダリ CISCO1.COM レルムに Linksys MTA のプロビジョニングに使用) する linksys-mta.tmpl テンプレート ファイルを関連付けます。
6. **[Add（追加）]** および **[Submit（送信）]** をクリックします。

詳細については、サービス クラスの設定を参照してください。

h) セカンダリ レルムのサービス クラスを追加します。この例では、CISCO2.COM。

次に例を示します。

1. **[Configuration（設定）] > [Class of Service（サービス クラス）]** の順に選択します。

2. [Add] をクリックします。[サービス クラスの追加（Add Class of Service）] ページが表示されます。
3. sa-mta として CISCO1.COM レルムの新しいサービス クラスの名前を入力します。
4. PacketCableMTA としてサービス クラス タイプを選択します。
5. プロパティ名ドロップダウンリストから*/cos/packetCableMTA/file*を選択し、(セカンダリ CISCO2.COM レルムに SA MTA のプロビジョニングに使用) する sa-mta.tmpl テンプレート ファイルを関連付けます。
6. **[Add（追加）]** および **[Submit（送信）]** をクリックします。

詳細については、サービス クラスの設定を参照してください。

ステップ 11 デバイスをオンラインして、それらをプロビジョニングします。プロビジョニング プロセスを説明する次の例を参照してください。

例 1

次の例では、Motorola SBV5120 をプロビジョニングする方法について説明します。

a) **sample-bronze-docsis** サービス クラスを使用するように設定して、デバイスのケーブル モデムの一部をプロビジョニングします。
b) MTA 部分をプロビジョニングするため、**[Devices（デバイス）] > [Manage Devices（デバイスの管理）]** ページに移動します。プロビジョニングする PacketCable デバイスを検索し、選択します。[Modify Device（デバイスの変更）] ページが表示されます。
c) ドメイン名を設定します。例では bacclab.cisco.com を使用します。
d) 登録済みサービス クラスに対応するドロップダウン リストから、**mot mta** を選択します。これは、手順 10-f で追加したサービス クラスです。
e) 登録されている DHCP 条件に対応するドロップダウンリストから **default** オプションを選択します。
f) [Submit] をクリックします。`

例 2

次の例は、Linksys CM2P2 をプロビジョニングする方法を示しています。

a) **sample-bronze-docsis** サービス クラスを使用するように設定して、設定してデバイスのケーブル モデム部分をプロビジョニングします。
b) MTA 部分をプロビジョニングするため、**[Devices（デバイス）] > [Manage Devices（デバイスの管理）]** ページに移動します。プロビジョニングする PacketCable デバイスを検索し、選択します。[Modify Device（デバイスの変更）] ページが表示されます。
c) ドメイン名を設定します。例では bacclab.cisco.com を使用します。
d) 登録済みのサービス クラスに対応するドロップダウン リストから、**llinksys-mta** を選択します。これは、手順 10-g で追加したサービス クラスです。
e) 登録されている DHCP 条件に対応するドロップダウンリストから **cisco1** オプションを選択します。これは、手順 10-a でセカンダリ CISCO1.COM レルムに追加した DHCP 条件に一致します。
f) [Submit] をクリックします。`

例 3

次の例は、SA WebStar DPX 2203 をプロビジョニングする方法を示しています。

a) **sample-bronze-docsis** サービス クラスを使用するように設定して、設定してデバイスのケーブル モデム部分をプロビジョニングします。
b) MTA 部分をプロビジョニングするため、**[Devices（デバイス）] > [Manage Devices（デバイスの管理）]** ページに移動します。プロビジョニングする PacketCable デバイスを検索し、選択します。[Modify Device（デバイスの変更）] ページが表示されます。
c) ドメイン名を設定します。例では bacclab.cisco.com を使用します。
d) 登録済みのサービス クラスに対応するドロップダウン リストから、**sa-mta** を選択します。これは、手順 10-h で追加したサービス クラスです。
e) 登録されている DHCP 条件に対応するドロップダウンリストから **cisco2** オプションを選択します。これは、手順 10-b でセカンダリ CISCO2.COM レルムに追加した DHCP 条件に一致します。
f) [Submit] をクリックします。`

ステップ 12 Ethereal トレースを使用して、複数のレルム サポートが動作しているか確認します。この手順で使用されているサンプル設定から、次のとおり KDC および DPE ログ ファイルの出力例を参照してください。

例 1

次の例では、プライマリ CISCO.COM レルムでプロビジョニングされた Motorola SBV 5120 MTA の、KDC と DPE ログ ファイルから例外を説明します。

KDC ログ サンプル出力 – Motorola MTA

```
INFO [Thread-4] 2007-02-07 07:56:21,133 (DHHelper.java:114) - Time to create DH key pair(ms): 48
 INFO [Thread-4] 2007-02-07 07:56:21,229 (DHHelper.java:114) - Time to create DH key pair(ms): 49

 INFO [Thread-4] 2007-02-07 07:56:21,287 (DHHelper.java:150) - Time to create shared secret: 57
ms.
 INFO [Thread-4] 2007-02-07 07:56:21,289 (PKAsReqMsg.java:104) - ##MTA-9a Unconfirmed AS Request:
 1133717956 Received from /10.10.1.2
 INFO [Thread-4] 2007-02-07 07:56:21,298 (KRBProperties.java:612) - Replacing property: 'minimum
 ps backoff' Old Value:'150' New Value: '150'
 INFO [Thread-4] 2007-02-07 07:56:21,324 (KDCMessageHandler.java:257) - AS-REQ contains PKINIT -
 QA Tag.
 INFO [Thread-4] 2007-02-07 07:56:21,325 (KDCMessageHandler.java:279) - PK Request from MTA
received. Client is MTA - QA Tag
 INFO [Thread-4] 2007-02-07 07:56:21,365 (KDCMessageHandler.java:208) - ##MTA-9b KDC Reply   AS-REP
 Sent to /10.10.1.2:1039 Time(ms): 290
 WARN [main] 2005-11-07 07:56:23,193 (KDC.java:113) - Statistics Report   ASREP's: 1
 INFO [main] 2005-11-07 07:56:23,195 (KDC.java:121) - /pktcbl/mtaAsRepSent: 10
 INFO [main] 2005-11-07 07:56:23,195 (KDC.java:121) - /pktcbl/DHKeygenTotalTime: 1043
 INFO [main] 2005-11-07 07:56:23,196 (KDC.java:121) - /pktcbl/mtaAsReqRecvd: 10
 INFO [main] 2005-11-07 07:56:23,197 (KDC.java:121) - /pktcbl/DHKeygenNumOps: 20
 INFO [main] 2005-11-07 07:56:23,197 (KDC.java:121) - /pktcbl/total: 60
```

DPE ログ サンプル出力 – Motorola MTA

```
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BAC-DPE-6-4178: Adding Replay Packet:  []
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BAC-PKTSNMP-6-0764: [System Description for MTA: <<HW_REV:
 1.0, VENDOR: Motorola Corporation, BOOTR: 8.1, SW_REV: SBV5120-2.9.0.1-SCM21-SHPC, MODEL: SBV5120>>]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BAC-PKTSNMP-6-0764: [##MTA-15 SNMPv3 INFORM Received From
 10.10.1.2.]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BAC-DPE-6-0688: Received key material update for device
 [1,6,01:11:82:61:5e:30]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BAC-PKTSNMP-6-0764: [##MTA-19 SNMPv3 SET Sent to 10.10.1.2]
```

```
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BAC-TFTP-6-0310: Finished handling [read] request from
[10.10.1.2:1190] for [bpr0106001182615e300001]
dpe.cisco.com: 2007 02 07 07:56:25 EST: %BAC-PKTSNMP-6-0764: [##MTA-25 SNMP Provisioning State
INFORM Received from 10.10.1.2.  Value: 1]
```

例 2

次の例では、セカンダリ CISCO1.COM レルムでプロビジョニングされた Linksys CM2P2 MTA の、KDC と DPE ログ ファイルから例外を説明します。

KDC ログ サンプル出力 – Linksys MTA

```
 INFO [Thread-8] 2007-02-07 08:00:10,664 (DHHelper.java:114) - Time to create DH key pair(ms): 49

 INFO [Thread-8] 2007-02-07 08:00:10,759 (DHHelper.java:114) - Time to create DH key pair(ms): 49

 INFO [Thread-8] 2007-02-07 08:00:10,817 (DHHelper.java:150) - Time to create shared secret: 57
ms.
 INFO [Thread-8] 2007-02-07 08:00:10,819 (PKAsReqMsg.java:104) - ##MTA-9a Unconfirmed AS Request:
 1391094112 Received from /10.10.1.5
 INFO [Thread-8] 2007-02-07 08:00:10,828 (KRBProperties.java:612) - Replacing property: 'minimum
 ps backoff' Old Value:'150' New Value: '150'
 INFO [Thread-8] 2007-02-07 08:00:10,860 (KDCMessageHandler.java:257) - AS-REQ contains PKINIT -
 QA Tag.
 INFO [Thread-8] 2007-02-07 08:00:10,862 (KDCMessageHandler.java:279) - PK Request from MTA
received. Client is MTA - QA Tag
 INFO [Thread-8] 2007-02-07 08:00:10,901 (KDCMessageHandler.java:208) - ##MTA-9b KDC Reply   AS-REP
 Sent to /10.10.1.5:3679 Time(ms): 296
 WARN [main] 2007-02-07 08:00:13,383 (KDC.java:113) - Statistics Report   ASREP's: 1
 INFO [main] 2007-02-07 08:00:13,384 (KDC.java:121) - /pktcbl/mtaAsRepSent: 11
 INFO [main] 2007-02-07 08:00:13,384 (KDC.java:121) - /pktcbl/DHKeygenTotalTime: 1141
```

DPE ログ サンプル出力 – Linksys MTA

```
dpe.cisco.com: 2007 02 07 08:00:10 EST: %BAC-DPE-6-4112: Adding Replay Packet:  []
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BAC-DPE-6-4178: Adding Replay Packet:  []
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BAC-PKTSNMP-6-0764: [System Description for MTA: Linksys
 Cable Modem with 2 Phone Ports (CM2P2) <<HW_REV: 2.0, VENDOR: Linksys, BOOTR: 2.1.6V, SW_REV:
2.0.3.3.11-1102, MODEL: CM2P2>>]
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BAC-PKTSNMP-6-0764: [##MTA-15 SNMPv3 INFORM Received From
 10.10.1.5.]
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BAC-DPE-6-0688: Received key material update for device
 [1,6,00:0f:68:f9:42:f6]
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BAC-PKTSNMP-6-0764: [##MTA-19 SNMPv3 SET Sent to 10.10.1.5]
dpe.cisco.com: 2007 02 07 08:00:18 EST: %BAC-TFTP-6-0310: Finished handling [read] request from
[10.10.1.5:1032] for [bpr0106000f68f942f60001]
dpe.cisco.com: 2007 02 07 08:00:18 EST: %BAC-PKTSNMP-6-0764: [##MTA-25 SNMP Provisioning State
INFORM Received from 10.10.1.5.  Value: 1]
```

例 3

次の例では、セカンダリ CISCO2.COM レルムでプロビジョニングされた SA WebStar DPX 2203 MTA の、KDC と DPE ログ ファイルから例外を説明します。

KDC ログ サンプル出力: SA MTA

```
 INFO [Thread-6] 2007-02-07 08:01:31,556 (DHHelper.java:114) - Time to create DH key pair(ms): 49

 INFO [Thread-6] 2007-02-07 08:01:31,652 (DHHelper.java:114) - Time to create DH key pair(ms): 50

 INFO [Thread-6] 2007-02-07 08:01:31,711 (DHHelper.java:150) - Time to create shared secret: 57
```

```
ms.
 INFO [Thread-6] 2007-02-07 08:01:31,715 (PKAsReqMsg.java:104) - ##MTA-9a Unconfirmed AS Request:
 575634000 Received from /10.10.1.50
 INFO [Thread-6] 2007-02-07 08:01:31,727 (KRBProperties.java:612) - Replacing property: 'minimum
 ps backoff' Old Value:'150' New Value: '150'
 INFO [Thread-6] 2007-02-07 08:01:31,752 (KDCMessageHandler.java:257) - AS-REQ contains PKINIT -
 QA Tag.
 INFO [Thread-6] 2007-02-07 08:01:31,753 (KDCMessageHandler.java:279) - PK Request from MTA
received. Client is MTA - QA Tag
 INFO [Thread-6] 2007-02-07 08:01:31,792 (KDCMessageHandler.java:208) - ##MTA-9b KDC Reply   AS-REP
 Sent to /10.10.1.50:3679 Time(ms): 292
 WARN [main] 2007-02-07 08:01:33,423 (KDC.java:113) - Statistics Report   ASREP's: 1
 INFO [main] 2007-02-07 08:01:33,424 (KDC.java:121) - /pktcbl/mtaAsRepSent: 12
 INFO [main] 2007-02-07 08:01:33,425 (KDC.java:121) - /pktcbl/DHKeygenTotalTime: 1240
 INFO [main] 2007-02-07 08:01:33,425 (KDC.java:121) - /pktcbl/mtaAsReqRecvd: 12
 INFO [main] 2007-02-07 08:01:33,426 (KDC.java:121) - /pktcbl/DHKeygenNumOps: 24
 INFO [main] 2007-02-07 08:01:33,426 (KDC.java:121) - /pktcbl/total: 72
```

DPE ログ サンプル出力: SA MTA

```
dpe.cisco.com: 2007 02 07 08:01:31 EST: %BAC-DPE-6-4112: Adding Replay Packet:  []
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BAC-DPE-6-4178: Adding Replay Packet:  []
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BAC-PKTSNMP-6-0764: [System Description for MTA: S-A
WebSTAR DPX2200 Series DOCSIS E-MTA Ethernet+USB (2)Lines VOIP <<HW_REV: 2.0, VENDOR: S-A, BOOTR:
 2.1.6b, SW_REV: v1.0.1r1133-0324, MODEL: DPX2203>>]
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BAC-PKTSNMP-6-0764: [##MTA-15 SNMPv3 INFORM Received From
 10.10.1.50.]
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BAC-DPE-6-0688: Received key material update for device
 [1,6,00:0f:24:d8:6e:f5]
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BAC-PKTSNMP-6-0764: [##MTA-19 SNMPv3 SET Sent to 10.10.1.50]
dpe.cisco.com: 2007 02 07 08:01:38 EST: %BAC-TFTP-6-0310: Finished handling [read] request from
[10.10.1.50:1037] for [bpr0106000f24d86ef50001]
dpe.cisco.com: 2007 02 07 08:01:39 EST: %BAC-PKTSNMP-6-0764: [##MTA-25 SNMP Provisioning State
INFORM Received from 10.10.1.50.  Value: 1]
```

複数のレルムでデバイスをプロビジョニングするためのテンプレートの作成

ここで説明されているテンプレート構文を使用して、特定のレルムでデバイスをプロビジョニングできます。ここに示す例では、Motorola SBV5120 MTA、Linksys CM2P2 MTA、SA WebStar DPX2203 MTA 特定のものです。プロビジョニングに使用されるそれぞれのテンプレートを次に示します。

（注） ネットワークで MTA の仕様に合わせてこれらのテンプレートを変更する必要があります。

Motorola MTA のプロビジョニングに使用するテンプレート

```
#
# Example PacketCable MTA template: mot-mta.tmpl
#
# Note that this template is specific to the TI 401 MTA.
# This template must be modified to the specifics of your MTA.
#
# First, the start marker.
#
```

```
option 254 1
#
# Enable MTA
#
option 11 .pktcMtaDevEnabled.0,INTEGER,true
#
# Set CMS FQDN for each endpoint on the MTA.
# NOTE: the indexes (9 and 10 here) will differ per manufacturer.
#
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.9,STRING,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,STRING,CMS.CISCO.COM
#
# Set the realm org name.  This MUST match that contained in the cert chain used by the
 device.
#
# "CableLabs, Inc."
option 11
.pktcMtaDevRealmTable.pktcMtaDevRealmEntry.pktcMtaDevRealmOrgName.'CISCO.COM',STRING,"CableLabs,
 Inc."
#
# Set the realm name and IPSec control for the CMS.
#
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsIpsecCtrl.'CMS.CISCO.COM',INTEGER,true
option 11
pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsKerbRealmName.'CMS.CISCO.COM',STRING,CISCO.COM
#
# Finally, the end marker.
#
option 254 255
```

Linksys MTA のプロビジョニングに使用するテンプレート

このテンプレートでは、レルムが CISCO1.COM に設定されていることに注意してください。

```
#
# Example PacketCable MTA template: linksys-mta.tmpl
#
# Note that this template is specific to the TI 401 MTA.
# This template must be modified to the specifics of your MTA.
#
# First, the start marker.
#
option 254 1
#
# Enable MTA
#
option 11 .pktcMtaDevEnabled.0,INTEGER,true
#
# Set CMS FQDN for each endpoint on the MTA.
# NOTE: the indexes (9 and 10 here) will differ per manufacturer.
#
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.9,STRING,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,STRING,CMS.CISCO.COM
#
# Set the realm org name.  This MUST match that contained in the cert chain used by the
 device.
#
# "CableLabs, Inc."
option 11
```

```
.pktcMtaDevRealmTable.pktcMtaDevRealmEntry.pktcMtaDevRealmOrgName.'CISCO1.COM',STRING,"CableLabs,
 Inc."
#
# Set the realm name and IPSec control for the CMS.
#
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsIpsecCtrl.'CMS.CISCO.COM',INTEGER,true
option 11
pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsKerbRealmName.'CMS.CISCO.COM',STRING,CISCO1.COM
#
# Finally, the end marker.
#
option 254 255
```

SA MTA のプロビジョニングに使用するテンプレート

テンプレートでは、レルムが CISCO2.COM に設定されていることに注意してください。

```
#
# Example PacketCable MTA template: sa-mta.tmpl
#
# Note that this template is specific to the TI 401 MTA.
# This template must be modified to the specifics of your MTA.
#
# First, the start marker.
#
option 254 1
#
# Enable MTA
#
option 11 .pktcMtaDevEnabled.0,INTEGER,true
#
# Set CMS FQDN for each endpoint on the MTA.
# NOTE: the indexes (9 and 10 here) will differ per manufacturer.
#
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.9,STRING,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,STRING,CMS.CISCO.COM
#
# Set the realm org name.  This MUST match that contained in the cert chain used by the
 device.
#
# "CableLabs, Inc."
option 11
.pktcMtaDevRealmTable.pktcMtaDevRealmEntry.pktcMtaDevRealmOrgName.'CISCO2.COM',STRING,"CableLabs,
 Inc."
#
# Set the realm name and IPSec control for the CMS.
#
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsIpsecCtrl.'CMS.CISCO.COM',INTEGER,true
option 11
pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsKerbRealmName.'CMS.CISCO.COM',STRING,CISCO2.COM
#
# Finally, the end marker.
#
option 254 255
```