Cisco Broadband Access Center アドミニストレータ ガイド

Release 4.0
December 2007

Text Part Number: OL-12296-01-J

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。

本書は、米国シスコシステムズ発行ドキュメントの参考和訳です。米国サイト掲載ドキュメントとの差異が生じる場合があるため、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

CONTENTS

このマニュアルについて

『*Cisco Broadband Access Center アドミニストレータ ガイド*』をご利用いただきありがとうございます。このマニュアルでは、Cisco Broadband Access Center（以下 BAC と表記）に関する概念と構成について説明します。

ここでは、このマニュアルの後続の章について概要を示し、この BAC リリースをサポートする関連資料の詳細情報を提供します。また、このマニュアルで使用されているスタイルと表記法についても説明します。

（注） このマニュアルは、P.xx の「製品マニュアル」およびP.xxi の「関連資料」に記載されているマニュアルと併せてお読みください。

ここでは、次の内容について説明します。

対象読者

システム管理者は、このマニュアルを使用して、ブロードバンド アクセスにおいて大規模なプロビジョニングを自動化する BAC を設定します。管理者は、次の内容について理解している必要があります。

- 基本的なネットワークの概念および専門用語
- ネットワーク管理
- ケーブル ネットワーク

マニュアルの構成

このマニュアルの主な内容は次のとおりです。

Broadband Access Center の概要	BAC について説明します。また、BAC リリースがサポートするテクノロジーと標準、その機能および利点についても説明します。
Broadband Access Center のアーキテクチャ	この BAC リリースに実装されているシステム アーキテクチャについて説明します。
設定のワークフロー	BAC を設定する際のワークフローを示します。
CPE プロビジョニングの概要	CPE プロビジョニングの概要を示し、BAC 内でサポートされている重要な概念について説明します。
設定テンプレートの管理	BAC がサポートしている設定テンプレートについて説明します。また、テンプレート ファイルを作成する方法についても説明します。
DOCSIS 設定	BAC DOCSIS 配備の使用方法について説明します。
PacketCable 音声設定	PacketCable 音声配備の使用方法について説明します。
CableHome の設定	ノンセキュア（DHCP）バージョンによる CableHome 配備の使用方法について説明します。
Broadband Access Center の管理	BAC の管理に役立つ各種サブコンポーネントについて説明します。
Broadband Access Center の監視	配備内の BAC サーバを監視する方法について説明します。
管理者のユーザ インターフェイスについて	管理者のユーザ インターフェイスから BAC にアクセスする方法について説明します。
管理者のユーザ インターフェイスの使用方法	管理アクティビティの実行方法について説明します。管理者ユーザ インターフェイスからのデバイス情報の検索および表示も含まれています。
Broadband Access Center の設定	管理者ユーザ インターフェイスからの設定アクティビティの実行方法について説明します。
サポートするツールと高度な概念	BAC の設定、保守、インストールの効率化、配備、使用に役立つ BAC ツールについて説明します。
データベースの管理	RDU データベースを管理および保守する方法について説明します。
Broadband Access Center のトラブルシューティング	PacketCable embedded Multimedia Terminal Adapter（eMTA; 組み込み型マルチメディア ターミナル アダプタ）のためのプロビジョニング プロセスをトラブルシューティングする方法について説明します。
アラートとエラー メッセージ	BAC アラート メッセージをリストで示し、説明します。
オプションのサポート	各テクノロジーのバージョンで BAC がサポートするテクノロジー固有のオプションをリストで示します。
PacketCable DHCP オプションと BAC プロパティのマッピング	PacketCable プロビジョニングで使用される PacketCable DHCP オプションと BAC プロパティのマッピングを確認します。
プロビジョニング API の使用例	共通のプロビジョニング API 使用例を紹介します。これを使用すると、典型的なサービス プロバイダーのワークフローをモデルにできます。

Broadband Access Center のプロビジョニングに関する FAQ	BAC の設定またはプロビジョニングについてのよくある質問を示します。
glossary	このマニュアルで使用されている用語と、説明されている技術に一般的に使用される用語を定義します。

表記法

このマニュアルは、次の表記法を使用しています。

項目	表記法
コマンドおよびキーワード	太字
ユーザが値を指定する変数	イタリック体
セッション情報およびシステム情報の表示出力	`screen` フォント
ユーザが入力する情報	太字の **`screen`** フォント
ユーザが入力する変数	イタリック体の *`screen`* フォント
メニュー項目およびボタン名	太字
本文中のメニュー項目の選択	**Option > Network Preferences**
表中のメニュー項目の選択	Option > Network Preferences

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

（注） 「注釈」です。役立つ情報や、このマニュアル以外の参照資料などを紹介しています。

ヒント *役立つヒント*の意味です。最適なアクションを示しています。

製品マニュアル

（注） 初版発行後、印刷物または電子マニュアルのアップデートを行う場合があります。マニュアルのアップデートについては、Cisco.com で確認してください。

表 1 では、今回の BAC リリースで使用できるマニュアルを示します。

表 1　製品マニュアル

マニュアル タイトル	ご利用形式
Release Notes for Cisco Broadband Access Center 4.0	• 製品 CD に収録されている PDF • Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_release_notes_list.html
Installation and Setup Guide for Cisco Broadband Access Center, Release 4.0	• 製品 CD に収録されている PDF • Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_installation_guides_list.html
Cisco Broadband Access Center Administrator Guide, Release 4.0	• 製品 CD に収録されている PDF • Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_maintenance_guides_list.html
Cisco Broadband Access Center DPE CLI Reference, Release 4.0	• 製品 CD に収録されている PDF • Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_command_reference_list.html

関連資料

（注） 初版発行後、印刷物または電子マニュアルのアップデートを行う場合があります。マニュアルのアップデートについては、Cisco.com で確認してください。

表 2 では、今回の BAC リリースで使用できる関連マニュアルを示します。

表 2　関連マニュアル

マニュアル タイトル	ご利用形式
Release Notes for Cisco Network Registrar 7.0	Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_release_notes_list.html
Installation Guide for Cisco Network Registrar, Release 7.0	Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_installation_guides_list.html
User Guide for Cisco Network Registrar, Release 7.0	Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps1982/products_user_guide_list.html
/docs ディレクトリ内の *CLIFrame.html*	Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_command_reference_list.html
Quick Start Guide for Cisco Network Registrar, Release 7.0	Cisco.com（次の URL を参照） http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_installation_guides_list.html

技術情報の入手方法およびサービス リクエストの発行

技術情報の入手、サービス リクエストの発行、その他の情報の収集に関する情報は、月刊の『*What's New in Cisco Product Documentation*』を参照してください。ここには、新規および改訂版のシスコの技術マニュアルもすべて記載されています。次の URL からアクセスできます。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Really Simple Syndication（RSS）フィードとして『*What's New in Cisco Product Documentation*』に登録し、リーダー アプリケーションを使用して、デスクトップに直接配信されるコンテンツを設定してください。RSS フィードは無料のサービスで、現在シスコでは RSS バージョン 2.0 をサポートしています。

■ 技術情報の入手方法およびサービス リクエストの発行

CHAPTER 1

Broadband Access Center の概要

この章では、Cisco Broadband Access Center リリース 4.0（以下、BAC）の概要について説明します。

この章は、次の項で構成されています。

BAC の概要

BAC は、ブロードバンド サービス プロバイダーのネットワークに存在する Customer Premises Equipment（CPE; 顧客宅内装置）をプロビジョニングおよび管理する作業を自動化します。

BAC は高性能の機能を備えているため、何百万ものデバイスが存在するネットワークなど、実質的にどのような規模のネットワークにも適合するように BAC を拡大、縮小できます。また、BAC は分散アーキテクチャと集中管理を備えているため、ハイ アベイラビリティを実現できます。

BAC は、サービス プロバイダーが急激に成長しても対応できるように設計されています。この製品の対象となるユーザは、ハイブリッド ファイバ ネットワークおよび同軸ケーブル ネットワーク上に IP データ、音声、動画を配置する必要があるブロードバンド サービス プロバイダー（マルチプル サービス オペレータを含む）、インターネット サービス プロバイダー、および音声サービス プロバイダーです。

BAC には、冗長性やフェールオーバーなどの重要な機能があります。BAC の動作方法を制御できるプロビジョニング Application Programming Interface（API; アプリケーション プログラミング インターフェイス）を利用することにより、新しい環境または既存の環境に BAC を統合することができます。プロビジョニング API を使用すると、BAC でのデバイス登録、デバイス構成の割り当て、および BAC プロビジョニング システム全体の設定が可能です。

このリリースの主な機能として、DOCSIS 3.0 仕様に準拠した CPE のプロビジョニングと管理のサポートがあります。DOCSIS 3.0 には主要なサブセットとして IP バージョン 6（IPv6）が含まれているため、このリリースは DHCPv6 と DNSv6 をサポートしています。

テクノロジーと機能

この項では、このリリースの BAC がサポートするテクノロジーと機能について説明します。

- サポート対象のテクノロジーと標準（P.1-2）
- サポート対象のデバイス（P.1-4）
- 機能と利点（P.1-4）

サポート対象のテクノロジーと標準

BAC では、ネットワークへプロビジョニング サービスを提供するために多くのテクノロジーのサポートが導入されました。次のようなテクノロジーがあります。

- DOCSIS 高速データ
- PacketCable 音声サービス（Secure ワークフローと Basic ワークフローの両方）
- ノンセキュア CableHome プロビジョニング

DOCSIS 高速データ

Data Over Cable Service Interface Specification（DOCSIS）は、ケーブル テレビのシステム ネットワーク上での高速データ配信に必要なケーブル モデムの機能を定義します。この機能により、MSO はインターネット常時接続を介してさまざまなサービスを提供できます。たとえば、ブロードバンド インターネット接続、テレフォニー、リアルタイムの対話型ゲーム、テレビ会議などのサービスを提供できます。

このリリースの BAC では、DOCSIS 1.0、1.1、2.0 のサポートの他に、DOCSIS 3.0 に準拠した CPE のプロビジョニングと管理を行います。DOCSIS 3.0 仕様は、第 3 世代の有線高速データ通信システム仕様を定義するもので、次の利点があります。

- IPv6 デバイスのプロビジョニング
- ネットワーク要素のアドレッシング機能の拡張
- チャネル ボンディングによるチャネル容量の増加
- ネットワーク セキュリティの強化
- マルチキャスト機能の拡大
- 新しいサービス オファリング

PacketCable 音声サービス

PacketCable 音声テクノロジーは、双方向ケーブル ネットワーク上で高度なリアルタイムのマルチメディア サービスの配信を可能にします。PacketCable は、ケーブル モデムがサポートするインフラストラクチャ上に構築され、IP テレフォニー、マルチメディア会議、対話型ゲーム、一般的なマルチメディア アプリケーションなど、幅広いマルチメディア サービスを可能にします。

PacketCable 音声テクノロジーにより、ブロードバンド ネットワークで、基本テレフォニー サービスや拡張テレフォニー サービスなどの付加的なサービスを提供できます。このようなサービスで、PacketCable は効率的かつコスト効果の高い方法です。

BAC は、PacketCable の Secure バリアントと Basic バリアントをサポートします。PacketCable Basic および PacketCable Secure は、Basic バリアントのセキュリティが簡略化されている点を除けば、ほとんど同じです。

（注）　現行の BAC では、PacketCable 仕様のバージョン 1.0、1.1、1.5 をサポートしています。

Euro-PacketCable サービスは、北米版 PacketCable 仕様に相当する欧州版です。両者の唯一の大きな違いは、Euro-PacketCable では異なる MIB が使用される点です。

CableHome

ノンセキュア CableHome 1.0 プロビジョニング（以下、ホーム ネットワーキング テクノロジー）は、既存の DOCSIS 標準上に構築され、住宅用ブロードバンド接続向け「プラグアンドプレイ」環境をサポートします。このような形態のホーム ネットワーキング テクノロジーには、CableHome をサポートする DOCSIS ホーム アクセス デバイスが含まれます。このデバイスはポータル サービスと呼ばれ、ホーム ネットワークのエントリ ポイントと見なされます。

サポート対象の標準

BAC サーバは、次の該当する Request for Comments（RFC）、プロトコル、標準、および Internet Engineering Task Force（IETF; インターネット技術特別調査委員会）草案に準拠しています。

- IPv6：RFC 2460（IPv6 仕様）、2461（近隣探索プロトコル）、2462（ステートレス アドレス自動設定）、2463（Internet Control Message Protocol（ICMP; インターネット制御メッセージ プロトコル））、3513（アドレッシング アーキテクチャ）に準拠しています。
- DHCPv6：RFC 3315（DHCPv6 仕様）、3633（IPv6 プレフィックス オプション）、3736（IPv6 用ステートレス DHCP サービス）、4014（Remote Authentication Dial-In User Service（RADIUS）リレー エージェント情報オプションの属性サブオプション）、4580（リレー エージェント加入者 ID オプション）、4649（リレー エージェント リモート ID オプション）、4704（DHCPv6 クライアント完全修飾ドメイン名（FQDN）オプション）に準拠しています。
- IPv4 と IPv6 の相互運用性：RFC 4038（IPv6 移行のアプリケーション）および 4472（IPv6 DNS に関する操作上の問題と考慮事項）に準拠しています。
- TFTP および ToD サーバ：RFC 868（タイム プロトコル）および 2349（TFTP ブロック サイズ オプション）に準拠しています。

さらに、BAC は次の該当する CableLabs 仕様および Comcast 仕様に準拠しています。

- DOCSIS 3.0 仕様
  - CM-SP-SECv3.0-I04-070518
  - CM-SP-PHY3.0-I04-070518
  - CM-SP-MULPIv3.0-I04-070518
  - CM-SP-OSSIv3.0-I03-070518
- DOCSIS 2.0 仕様 CM-SP-RFI2.0-I12-071206
- DOCSIS L2VPN 仕様 CM-SP-L2VPN-I06-071206
- PacketCable MTA デバイス プロビジョニング仕様 PKT-SP-PROV1.5-I03-070412
- CableHome CH-SP-CH1.0-I05-030801
- COMCAST-SP-RNG-200-ProvOSS-I04-070102
- OpenCable 仕様 OC-SP-HOST2.0-CFR-I13-070323

サポート対象のデバイス

このリリースの BAC では、次のプロビジョニングと管理をサポートします。

- IPv6 デバイス。次のものがあります。
  - DOCSIS 3.0 準拠のケーブル モデム
  - コンピュータ
  - Set-top box（STB; セットトップ ボックス）
- ビデオ STB（特に進化途上の OpenCable アプリケーション プラットフォームに基づく RNG-200 STB）
- embedded Service/Application Functional Entities（eSAFE）デバイスのバリアント。混在 IP モードの PacketCable Multimedia Terminal Adapters（MTA; マルチメディア ターミナル アダプタ）などがあります。混在 IP モード MTA は、IPv6 組み込みケーブル モデムと IPv4 eMTA で構成される eSAFE デバイスです。このクラスのデバイスでは、ケーブル モデムにパケット テレフォニー、ホーム ネットワーキング、動画などの付加的機能が組み込まれています。

BAC は次のデバイス タイプのプロビジョニングを行います。

- DOCSIS 1.0、1.1、2.0 準拠のケーブル モデムと STB
- PacketCable バージョン 1.x 準拠の embedded Multimedia Terminal Adapter（eMTA; 組み込み型マルチメディア ターミナル アダプタ）
- CableHome 1.0 準拠のデバイス
- コンピュータ

機能と利点

BAC により、Multiple Service Operator（MSO; マルチプル サービス オペレータ）は、急速に変化するケーブル データ通信サービスへの需要に対応できます。BAC を使用して、次の利点を実現できます。

- 大幅なスケーラビリティ。BAC の Regional Distribution Unit（RDU）は、最大 6000 万台のデバイスをサポートできます。また、1 つのプロビジョニング グループは 200 万台のデバイスをサポートでき、そのうち 50 万台を Secure PacketCable デバイスにすることができます。
- バックエンド システムとの容易な統合。この統合には、次のような BAC メカニズムが使用されます。
  - BAC Java API。すべてのプロビジョニング操作および管理操作を実行するときに使用できます。
  - BAC パブリッシング拡張。RDU データを別のデータベースに書き込むときに便利です。
  - SNMP エージェント。BAC の監視に関する統合を簡素化します。
  - DPE コマンドライン インターフェイス（CLI）。「サービス」インターフェイスを介して DPE を要件に合せて設定できます。また、CLI を使用してコマンドをコピー アンド ペーストするとローカル構成を簡素化できます。
- 管理の向上
  - プロパティ階層のプロビジョニング グループ プロパティ。デバイスのプロビジョニング グループのプロパティが含まれたことで、BAC プロパティ階層の柔軟性が向上します。
  - プロビジョニング グループ機能。配備内のプロビジョニング グループに必要なデバイス タイプ サポートを制御できます。

- セキュリティの強化
  - ユーザ設定可能な IP アドレスとポート。マルチパス設定、マルチインターフェイス バインディング、およびファイアウォール機能が提供されます。
  - 拡張 CMTS MIC 設定内容の DOCSIS 3.0。BAC が高度なハッシュ技術を使用してケーブルモデム設定ファイルの不正な変更や破壊を検出します。
  - パスワード ポリシー。管理者のユーザ インターフェイスから RDU にアクセスできます。管理者のユーザ インターフェイスへのログインに使用するパスワードは、必ず 8 文字以上にします。
  - HTTP over SSL（HTTPS）。セキュア SSL 接続を使用して管理者のユーザ インターフェイスにアクセスできます。
- トラブルシューティングと診断の強化
  - デバイス トラブルシューティング。デバイスのトラブルシューティング用 ID を使用して、デバイスと BAC サーバとのインタラクションの詳細なレコードを提供します。この機能を使用すると、MAC アドレスまたは DHCP Unique Identifier（DUID）で特定される 1 台のデバイスに焦点を絞り、その診断情報を使用してより詳細に分析できます。
  - 診断スクリプトを使用したサーバ トラブルシューティング。BAC サーバに関するパフォーマンス統計情報を（具体的なタイプの統計情報まで）収集します。また、このリリースでは、サーバおよびシステムの設定データを収集するための多くのスクリプトを提供します。このデータはサポートで必要となる場合があります。追加のスクリプトを使用して、サポート用に診断データを組み込むことができます。

CHAPTER 2

Broadband Access Center のアーキテクチャ

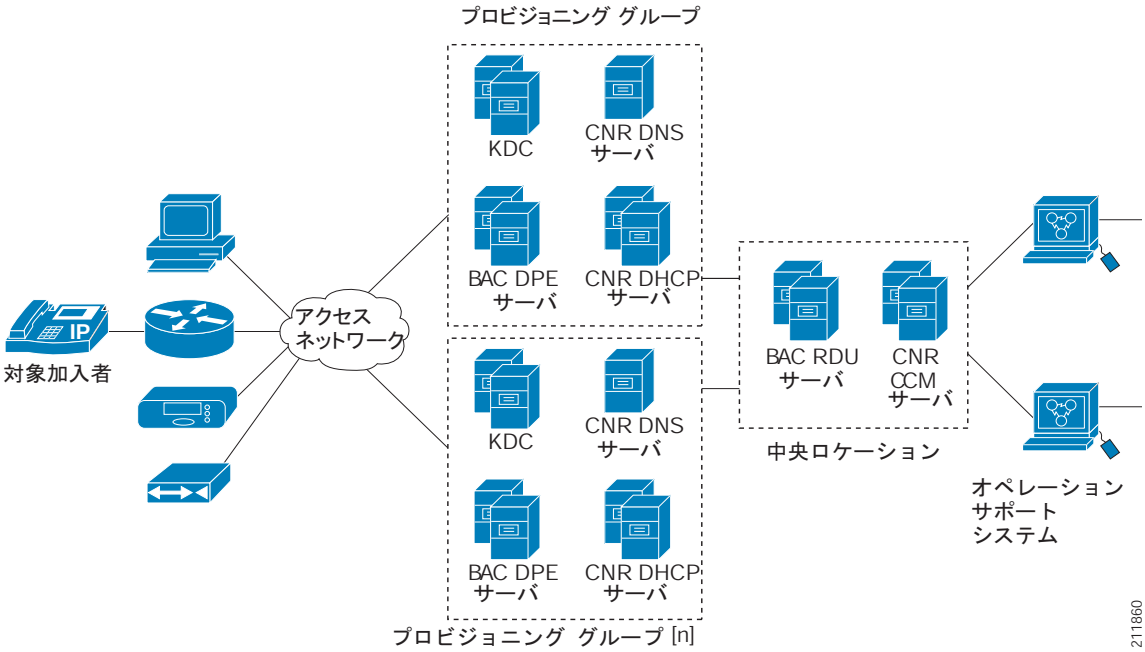
この章では、この Cisco Broadband Access Center（BAC）リリースに実装されているシステム アーキテクチャについて説明します。

この章は、次の項で構成されています。

配備

図 2-1 は、BAC ネットワークにおける一般的な完全冗長の配備を示しています。

図 2-1　BAC を使用した配備



アーキテクチャ

ここでは、次に示すコンポーネントから成る BAC の基本アーキテクチャについて説明します。

- Regional Distribution Unit（RDU）。次の機能を提供します。
  - BAC システムの権限あるデータ格納
  - Application Programming Interface（API; アプリケーション プログラミング インターフェイス）の要求を処理するためのサポート
  - システム全体のステータスおよび状態の監視

  詳細については、P.2-3 の「Regional Distribution Unit」を参照してください。
- Device Provisioning Engine（DPE）。次の機能を提供します。
  - 顧客宅内装置（CPE）とのインターフェイス
  - 構成キャッシュ
  - RDU および他の DPE から独立した操作
  - PacketCable プロビジョニング サービス
  - 構成用の IOS ライクなコマンドライン インターフェイス（CLI）

  詳細については、P.2-6 の「Device Provisioning Engine」を参照してください。
- クライアントからシステムの機能をすべて制御できるようにする BAC API。
- Cisco Network Registrar サーバ。次の機能を提供します。
  - Dynamic Host Configuration Protocol（DHCP）
  - Domain Name System（DNS; ドメイン ネーム システム）

  詳細については、P.2-13 の「Network Registrar」を参照してください。
- プロビジョニング グループ。次の機能を提供します。
  - 冗長クラスタ内の Network Registrar サーバおよび DPE の論理的なグループ化
  - 冗長性とスケーラビリティ

  詳細については、P.2-16 の「プロビジョニング グループ」を参照してください。
- Kerberos サーバ。PacketCable Multimedia Terminal Adapter（MTA; マルチメディア ターミナル アダプタ）を認証します。詳細については、P.2-14 の「Key Distribution Center」を参照してください。
- BAC プロセス ウォッチドッグ。次の機能を提供します。
  - すべての重要な BAC プロセスに対する管理のための監視
  - プロセス自動再開機能
  - BAC コンポーネント プロセスを開始および中止する機能

  詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。
- SNMP エージェント。次の機能を提供します。
  - サードパーティの管理システム
  - SNMP バージョン v2
  - SNMP 通知

  詳細については、P.9-5 の「SNMP エージェント」を参照してください。
- 管理者のユーザ インターフェイス。次の機能をサポートします。
  - デバイスの追加、削除、修正、および検索
  - グローバル デフォルトの設定およびカスタム プロパティの定義

  詳細については、P.2-15 の「管理者のユーザ インターフェイス」を参照してください。

Regional Distribution Unit

RDU は BAC プロビジョニング システムの主要サーバです。RDU は Solaris オペレーティング システムを実行しているサーバにインストールする必要があります。

RDU には次の機能があります。

- デバイス構成生成の管理
- デバイス構成の生成、およびキャッシングのための DPE への配信
- デバイス構成を最新の状態に保つための DPE との同期
- すべての BAC 機能に対する API 要求の処理
- BAC システムの管理

RDU は、拡張性のあるアーキテクチャにより新しい技術とサービスの追加をサポートします。

現行の BAC では、1 回のインストールで 1 つの RDU がサポートされます。フェールオーバーをサポートする場合は、Veritas または Sun のクラスタリング ソフトウェアを使用することをお勧めします。フェールオーバーのセットアップでは、RAID（冗長ディスクアレイ）の共有ストレージを使用することをお勧めします。

以降の項では、次の RDU の概念について説明します。

- デバイス構成の生成（P.2-3）
- サービス レベル選択（P.2-4）

デバイス構成の生成

デバイスはブート時に、BAC に構成を要求します。この構成がデバイスのサービス レベルを決定します。この処理中、DHCP サーバは RDU にデバイスの構成を作成するように要求します。RDU は構成を生成して、そのデバイスが属するプロビジョニング グループにサービスを提供するすべての DPE に転送します。これで、DPE は RDU にアクセスしなくてもデバイスに構成を提供できるようになります。

デバイス構成には、ユーザが必要とする次のプロビジョニング情報を含めることができます。

- DHCP IP アドレス選択
- 帯域幅
- データ レート
- フロー制御
- 通信速度
- サービス レベル

構成には、任意のデバイスの DHCP 構成と TFTP ファイルを含めることができます。プロビジョニングされていないデバイスをインストールしてブートすると、デフォルトのテクノロジー固有の構成が割り当てられます。BAC がサポートするテクノロジーごとにデフォルト構成を変更できます。

RDU は、次の場合にデバイスの構成を再生成します。

- デバイスのサービス クラス変更など、特定のプロビジョニング API コールが実行された場合。
- 構成の検証が失敗した場合。デバイスから送信された DHCP 要求の特定のパラメータが最初の要求パラメータと異なる場合などに発生します。
- DPE がキャッシュへの再読み込みを行った場合。

RDU がデバイスの構成を再生成するたびに、更新された構成が該当する DPE に転送されます。

サービス レベル選択

サービス レベル選択の拡張ポイントは、RDU がデバイスの構成を生成するときに使用する DHCP 基準とサービス クラスを決定します。RDU は、この情報をデバイスごとにデータベースに保存します。

RDU がデバイス構成の生成に使用する DHCP 基準とサービス クラスは、デバイスに付与されているアクセス権のタイプに基づきます。デバイスのアクセス権には次の 3 つのタイプがあります。

- デフォルト：デフォルト アクセス権を付与されているデバイスの場合、BAC はデバイス タイプに割り当てられているデフォルトのサービス クラスと DHCP 基準を使用します。
- 無差別：無差別アクセス権を付与されているデバイスの場合、BAC はそのデバイスが背後にあるリレー エージェントからサービス クラスと DHCP 基準を取得します。
- 登録済み：登録済みアクセス権を付与されているデバイスの場合、BAC は RDU データベース内にある、デバイスの登録済みサービス クラスと DHCP 基準を使用します。

デバイス タイプごとに必ず 1 つのデフォルト拡張が存在している必要があります。

Configuration > Defaults のデフォルト ページを使用して特定のテクノロジーのサービス レベル選択の拡張ポイントを入力できます。詳細については、P.13-7 の「デフォルトの設定」を参照してください。デフォルトでは、これらのプロパティにゼロまたは組み込み拡張の 1 つが入力されます。

注意　独自のカスタム拡張をインストールしている場合を除き、これらの拡張を修正しないでください。

デバイスは、DHCP 基準とサービス クラスのセットを 1 つ受け付けるように登録されていても、実際には 2 つ目のセットを選択する場合があります。構成生成の拡張は、選択された DHCP 基準とサービス クラスを検索して使用します。

サービス レベル選択拡張は、デバイスに対して指定された特定のルールに基づいて 2 つ目のサービス クラスと DHCP 基準を選択します。たとえば、特定のプロビジョニング グループ内のデバイスについて、特定のサービス クラスと DHCP 基準を割り当てるには、そのデバイスをブートするように指定する場合があります。

拡張は、DHCP 基準とサービス クラスの特定のセットがデバイスのプロビジョニングに選択される理由となる情報を返します。これらの理由は、View Device Details ページの管理者のユーザ インターフェイスで確認できます。

表 2-1 に、これらの理由とその場合に付与されるアクセス権のタイプを示します。

表 2-1　サービス レベル選択拡張がデバイスのアクセス権を決定した理由

理由コード	説明	付与されるデバイス アクセス権のタイプ		
		デフォルト	無差別	登録済み
NOT_BEHIND_REQUIRED_DEVICE	必要なリレー エージェントの背後にデバイスがありません。	✓		
NOT_IN_REQUIRED_PROV_GROUP	必要なプロビジョニング グループにデバイスが含まれていません。	✓		
NOT_REGISTERED	デバイスが登録されていません。	✓		
PROMISCUOUS_ACCESS_ENABLED	リレー エージェントに対して無差別モードのアクセス権がイネーブルになっています。		✓	
REGISTERED	デバイスが登録されています。			✓
RELAY_NOT_IN_REQUIRED_PROV_GROUP	必要なプロビジョニング グループにリレー エージェントが含まれていません。	✓		
RELAY_NOT_REGISTERED	リレー エージェントが登録されていません。	✓		

（注） これらの理由のほとんどは、登録済みまたは無差別のアクセス権を付与するための要件違反を示しており、結果としてデフォルト アクセス権が付与されています。

Device Provisioning Engine

Device Provisioning Engine（DPE）は、CPE と通信してプロビジョニング機能および管理機能を実行します。

RDU は、DHCP 命令とデバイス構成ファイルを生成し、それらを関連する DPE サーバに配信します。DPE は DHCP 命令とデバイス構成ファイルをキャッシュします。次に、DHCP 命令は Network Registrar 拡張とのインタラクションで使用され、構成ファイルは TFTP サーバを介してデバイスに配信されます。

BAC は複数の DPE をサポートします。複数の DPE を使用して、冗長性とスケーラビリティを確保できます。

DPE は、デバイスへの構成ファイルの提供を含む、すべての構成要求を処理します。DPE は Network Registrar DHCP サーバと統合され、各デバイスの IP アドレスの割り当てを制御します。複数の DPE が 1 つの DHCP サーバと通信できます。

DPE は次に示すアクティビティを管理します。

- 最新の構成を取得してキャッシュするために RDU と同期する。
- 最終段階のデバイス構成を生成する（DOCSIS タイムスタンプなど）。
- DHCP サーバに DHCP メッセージ交換を制御する命令を提供する。
- TFTP を介して構成ファイルを配信する。
- Network Registrar と統合する。
- 音声テクノロジー サービスをプロビジョニングする。

DPE を Solaris オペレーティング・システムを実行するサーバにインストールする必要があります。CLI から DPE の設定と管理を行います。CLI には、ローカルで、または Telnet を介してリモートでアクセスできます。DPE がサポートする CLI コマンドの詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

（注） インストール中に、各 DPE に対して次の項目を設定する必要があります。

- DPE が属するプロビジョニング グループの名前。この名前によって、DPE がサービスを提供するデバイスの論理グループが決まります。
- RDU の IP アドレスとポート番号。

DPE に関する重要な情報については、次の項を参照してください。

- DPE のライセンシング（P.2-7）
- TACACS+ および DPE 認証（P.2-7）
- DPE と RDU 間の同期（P.2-8）
- TFTP サーバ（P.2-10）
- ToD サーバ（P.2-11）

また、P.2-16 の「プロビジョニングの概念」の情報も十分に理解しておいてください。

DPE のライセンシング

ライセンシングにより、ご使用になれる DPE（ノード）の数が管理されます。お持ちのライセンスより多くの DPE をインストールしようとすると、新しい DPE は RDU に登録されず、拒否されます。ライセンスされている既存の DPE は、オンラインのままになります。

（注） ライセンシングの目的上、登録済みの DPE は 1 つのノードと見なされます。

ライセンスの追加または評価ライセンスの延長を行った場合、または評価ライセンスが満了した場合、その変更はただちに有効になります。

RDU データベースから登録済みの DPE を削除すると、ライセンスは解放されます。DPE は RDU に自動的に登録されるため、ライセンスを開放する場合は DPE をオフラインにする必要があります。次に、管理者のユーザ インターフェイスまたは API を介して、DPE を RDU データベースから削除します。

削除した DPE は、属するすべてのプロビジョニング グループから削除されます。すべての Network Registrar 拡張に、その DPE が使用できなくなったことが通知されます。そのため、以前に削除された DPE が再度登録されると、再度ライセンスされたと見なされ、RDU から再度削除されるかライセンスが満了するまで、その状態が続きます。

RDU を使用してライセンスされていない DPE は、管理者のユーザ インターフェイスに表示されません。ライセンスの状態を判断するには、DPE と RDU のログファイルを調べる必要があります（*dpe.log* と *rdu.log*）。

（注） 特定のライセンスを介してイネーブルにした機能は、対応するライセンスがシステムから削除されても、引き続き動作します。

ライセンシングの詳細については、P.13-23 の「ライセンスの管理」を参照してください。

TACACS+ および DPE 認証

TACACS+ は TCP ベースのプロトコルで、多くのネットワーク デバイスを対象とする中央集中型のアクセスと DPE CLI のユーザ認証をサポートします。

TACACS+ を使用して、DPE は複数のユーザをサポートできます。各ユーザ名とログイン パスワードおよびイネーブル パスワードは、TACACS+ サーバで設定されます。TACACS+ は、TACACS+ クライアント / サーバ プロトコルを実装するために使用されます（ASCII ログインのみ）。

TACACS+ 特権レベル

TACACS+ サーバは、TACACS+ プロトコルを使用して DPE にログインするユーザを認証します。TACACS+ クライアントは、そのユーザに設定される特定のサービス レベルを指定します。

表 2-2 は、DPE ユーザのアクセスを許可するために使用される 2 つのサービス レベルを示します。

表 2-2　TACACS+ サービス レベル

モード	説明
ログイン	`router>` プロンプトでのユーザレベル コマンド
イネーブル	`router#` プロンプトでのイネーブルレベル コマンド

TACACS+ クライアント設定

TACACS+ は、CLI を使用して設定される複数のプロパティを使用します。TACACS+ に関するコマンドの詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

TACACS+ をイネーブルにする場合、管理者はすべての TACACS+ サーバの IP アドレスまたは Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を、デフォルト値以外に指定する必要があります。

次の設定についても、該当する場合はデフォルト値を使用して指定できます。

- 各 TACACS+ サーバの共有秘密鍵。この鍵を使用して、DPE と TACACS+ サーバの間のデータを暗号化できます。特定の TACACS+ サーバの共有秘密情報を省略することを選択した場合、TACACS+ メッセージ暗号化は使用されません。
- TACACS+ サーバのタイムアウト値。この値を使用して、TACACS+ サーバがプロトコル要求に応答するのを TACACS+ クライアントが待機する最長時間を指定できます。
- TACACS+ サーバのリトライ回数。この値を使用して、TACACS+ クライアントが TACACS+ サーバとの有効なプロトコル交換を試行する回数を指定できます。

DPE と RDU 間の同期

DPE と RDU 間の同期とは、RDU との整合を取るために、DPE キャッシュを自動的に更新するプロセスです。DPE キャッシュは、デバイスの構成を含む構成キャッシュと、デバイスに必要なファイルを含むファイル キャッシュから構成されます。

通常、RDU は、構成の更新が含まれるイベントを生成し、関係するすべての DPE に送信して DPE を最新の状態に保ちます。同期が必要になるのは、接続の切断によって DPE 側でいくつかのイベントが欠落した場合です。切断の原因としては、ネットワークの問題、管理目的での DPE サーバのダウン、または障害などが考えられます。

また、同期は、RDU データベースをバックアップから復元するという特殊なケースでも使用されます。このケースでは、RDU との整合を取るために、DPE キャッシュのデータベースを古い状態に戻す必要があります。

RDU と DPE 間の同期プロセスは自動的に実行されるため、管理操作は必要ありません。同期プロセスの実行中でも、DPE は CPE に対してプロビジョニング操作や管理操作をすべて実行できます。

同期プロセス

DPE は RDU との接続を確立するたびに同期プロセスをトリガーします。

DPE は最初に起動したときに、RDU への接続を確立し、RDU に登録して構成変更の更新を受信します。次に、DPE と RDU が、ハートビート メッセージの交換を使用して、接続を監視します。DPE は、RDU への接続が切断されたと判断すると、接続の再確立を自動的に試みます。この試行は、成功するまでバックオフのリトライ間隔で続行されます。

RDU も、接続の切断を検出すると、DPE へのイベントの送信を停止します。接続の切断によって RDU からの更新イベントが DPE 側で欠落する場合があるため、DPE は、RDU との接続を確立するたびに同期を実行します。

一般的な DPE の状態

同期プロセス中、DPE は次の状態になります。

1. Registering：RDU との接続を確立して登録するプロセスの間、DPE は *Registering* 状態になっています。
2. Synchronizing：DPE は、必要な構成のグループを RDU に要求します。このプロセスの間、DPE は、ストレージ内の構成のうち、整合の取れていない（バージョン番号が間違っている）もの、欠落しているもの、および削除するものを判別し、必要であればキャッシュ内の構成を更新します。また、DPE は TFTP サーバに配信可能なキャッシュ内のファイルを同期します。DPE では、RDU が構成要求によって過負荷にならないようにするため、一度に 1 つのバッチだけを中央サーバに送信します。
3. Ready：DPE は最新の状態であり、RDU と完全に同期が取れています。通常 DPE はこの状態にあります。

表 2-3 に、DPE に発生し得るその他の状態を示します。

表 2-3　関連する DPE の状態

状態	説明
Initializing	起動中です。
Shutting Down	停止処理中です。
Down	Network Registrar 拡張ポイントからのクエリーに応答しません。
Ready Overloaded	DPE が実行されているシステムに過大な負荷がかかっている点を除き、*Ready* と同様です。

（注） DPE の状態に関係なく、デバイス構成要求、TFTP 要求、および ToD 要求に引き続きサービスが提供されます。

DPE 状態は次の方法で表示できます。

- 管理者のユーザ インターフェイスから表示します。P.12-23 の「Device Provisioning Engine の表示」を参照してください。
- **show dpe** コマンドを使用して DPE CLI から表示します。『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

TFTP サーバ

統合された TFTP サーバは、デバイスおよびデバイス以外のエンティティから、DOCSIS 構成ファイルを含む、ファイル要求を受信します。その後、このサーバはファイルを要求元のエンティティに送信します。

TFTP サーバは、ローカル ファイルシステム アクセスに使用されるホーム ディレクトリにあります。ローカル ファイルは、*BPR_DATA/dpe/tftp* ディレクトリに格納されています。このリリースでは、すべての配信可能な TFTP ファイルは、事前に DPE にキャッシュされます。つまり、DPE は常にシステム内のすべてのファイルが使用された最新の状態になっています。

(注) DPE での TFTP サービスには、要件に合せてサービスを設定できる機能があります。

デフォルトでは、TFTP サーバは TFTP 読み取りに対してキャッシュしか検索しません。ただし、DPE コマンドラインから **service tftp** *1..1* **allow-read-access** コマンドを実行すると、TFTP サーバはキャッシュの前にローカル ファイル システムを検索します。ファイルがローカル ファイル システムにある場合は、そこからファイルを読み取ります。ファイルがない場合、TFTP サーバはキャッシュを検索します。ファイルがキャッシュにある場合、サーバはそのファイルを使用し、キャッシュにない場合はエラーを返します。

ローカル ファイル システムからの読み取りアクセスをイネーブルにできる場合、ディレクトリ構造読み取り要求は、ローカル ファイル システムからだけ許可されます。

(注) すべての TFTP ファイルに一意の名前を指定します。ファイルを大文字と小文字を使用して区別しないようにしてください。DPE は、ローカル ディレクトリまたはキャッシュでファイルを検索するとき、すべてのファイル名を小文字に変換するため、ファイル名の文字は重要です。

IPv4 または IPv6 を使用する TFTP 転送を指定するには、DPE コマンドラインから **service tftp** *1..1* **ipv4 | ipv6 enabled true** コマンドを使用します。また、転送のブロックサイズを指定するには、***service tftp*** *1..1* ***ipv4 | ipv6 blocksize*** コマンドを使用します。blocksize オプションには、データ オクテットの数を指定します。これにより、クライアントとサーバが、ネットワーク メディアにより適したブロック サイズをネゴシエートできます。blocksize をイネーブルにすると、TFTP サービスは、要求されたブロック サイズが指定された上限と下限の間に収まっていれば、転送に使用します。詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

TFTP サービスは、TFTPv4 と TFTPv6 について、処理された TFTP パケット数の統計情報を保持しています。これらの統計情報は、Device Details ページの管理者のユーザ インターフェイスで表示できます。詳細については、P.12-10 の「デバイスの詳細の表示」を参照してください。

ToD サーバ

BAC の統合された Time of Day（ToD）サーバは、RFC 868 の高性能 UDP を実装します。

（注）DPE での ToD サービスには、要件に合せてサービスを設定できる機能があります。

ToD サービスをイネーブルにして IPv4 または IPv6 をサポートするには、DPE コマンドラインから **service tod** *1..1* **enabled true** コマンドを使用します。デフォルトでは、ToD サービスは DPE でディセーブルになっています。

このプロトコルを DPE に設定するときには、プロビジョニング用に設定したインターフェイスにだけ ToD サービスはバインドされることに注意してください。ToD サービスの設定の詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

ToD サービスは、ToDv4 と ToDv6 について、処理された ToD パケット数の統計情報を保持しています。これらの統計情報は、Device Details ページの管理者のユーザ インターフェイスで表示できます。詳細については、P.12-10 の「デバイスの詳細の表示」を参照してください。

DOCSIS 共有秘密情報

BAC では、Cable Modem Termination System（CMTS; ケーブル モデム ターミネーション システム）ごとに異なる DOCSIS Shared Secret（DSS; DOCSIS 共有秘密情報）を定義できます。このように定義すると、共有秘密情報が侵害されても影響を受ける CMTS の数が限られ、配備におけるすべての CMTS が対象となりません。

DSS は DPE ごとに設定できますが、プロビジョニング グループ単位で設定する必要があります。また、そのプロビジョニング グループで CMTS に設定されている内容と一致させる必要があります。

注意　1 つのプロビジョニング グループ内に複数の DSS を設定すると、場合によって CMTS のパフォーマンスが低下する可能性があります。ただし、実質的に BAC には影響がありません。

共有秘密情報はクリア テキスト文字列または IOS 暗号化文字列で入力できます。クリア テキストで入力する場合、DSS は IOS バージョン 12.2BC に適合するように暗号化されます。

また、管理者のユーザ インターフェイスまたは API を使用して、RDU から DSS を設定することもできます。この場合、DSS はクリア テキストで入力され、RDU に格納されて、すべての DPE に渡されます。そのため、このように入力された DSS は、DPE に保存される前に暗号化されます。

CLI から ***dpe docsis shared-secret*** コマンドを使用して直接 DSS を DPE で設定する場合、この設定は RDU から設定される内容よりも優先されます。

DOCSIS 共有秘密情報のリセット

DSS のセキュリティが侵害された場合、または単に管理目的で共有秘密情報を変更する場合は DSS をリセットできます。

DSS をリセットするには、CMTS CLI から **show running-config** コマンドを実行し、表示された設定から DOCSIS 共有秘密情報を DPE 設定にコピー アンド ペーストします。この方法で、Cisco CMTS に入力した設定を DPE CLI にコピーできます。

（注） 上記で説明されたように共有秘密情報を変更するには、CMTS は 12.2BC より新しいソフトウェアバージョンで実行されている必要があります。

DSS を変更するには、次の手順に従います。

ステップ 1 リセットする DOCSIS 共有秘密情報を持つプロビジョニング グループを確認します。

ステップ 2 そのプロビジョニング グループに関連付けられた DPE と CMTS のリストを調べます。

ステップ 3 CMTS 上でプライマリ DSS を変更します。

ステップ 4 CMTS 上で侵害された DSS をセカンダリ DSS に変更します。すべての DOCSIS 設定ファイルが新しい DSS を使用するように正しく変更されるまで、ケーブル モデムで登録を実行できるようにする必要があります。

ステップ 5 影響を受けた DPE を判別し、それぞれ DSS を変更します。

ステップ 6 DOCSIS 設定ファイルが新しい DSS を使用していることを確認してから、侵害されたセカンダリ共有秘密情報を CMTS 設定から削除します。

Network Registrar

Network Registrar は、BAC で DHCP および DNS 機能を提供します。Network Registrar の DHCP 拡張ポイントは、BAC を Network Registrar と統合します。これらの拡張を使用して、BAC は DHCP 要求の内容を調べてデバイス タイプを検出し、その構成に従って内容を操作し、プロビジョニングするデバイスのカスタマイズされた構成を配信します。

Network Registrar の詳細については、『*User Guide for Cisco Network Registrar 7.0*』、*/docs* ディレクトリの *CLIFrame.html*、および『*Installation Guide for Cisco Network Registrar, 7.0*』を参照してください。

DHCP

DHCP サーバは、IP ネットワーク上で IP アドレスを設定するプロセスを自動化します。このプロトコルは、デバイスをネットワークに接続するときにシステム管理者が行う多くの機能を実行します。DHCP はネットワーク ポリシーの決定を自動的に管理するため、手動の設定が不要になります。これにより、ネットワーク デバイスの設定の柔軟性とモビリティが高まり、管理が容易になります。

このリリースの BAC は、IPv6 用の DHCP（DHCPv6 とも呼ばれる）をサポートします。DHCPv6 を使用すると、DHCP サーバは、拡張を介して設定パラメータを IPv6 ホストに配信できるようになります。IPv6 ホストは、デフォルトでステートレス自動設定を使用します。これは、IPv6 ホストがローカル IPv6 ルータを使用して独自のアドレスを設定できるようにするものです。DHCPv6 とは、このステートフル自動設定オプションのことで、サーバがホストに設定情報を設定する技術の 1 つです。

DHCPv6 には次の利点があります。

- IPv6 アドレスによるアドレッシング機能の拡張
- ステートフル自動設定プロトコルを使用した容易なネットワーク管理
- オプションと拡張のサポート向上
- リレー エージェント機能
- 1 つのインターフェイスへの複数アドレスの割り当て

DHCPv4 と DHCPv6 の比較

DHCPv6 は、DHCPv4 と同様にクライアント / サーバ モデルを使用します。DHCP サーバと DHCP クライアントは、IP アドレスの要求、オファー、およびリースを行うために一連のメッセージを交わします。DHCPv4 とは異なり、DHCPv6 は、一括して会話を行う場合、ブロードキャスト メッセージではなく、ユニキャスト メッセージとマルチキャスト メッセージの組み合せを使用します。

この他にも DHCPv4 と DHCPv6 には次のような違いがあります。

- DHCPv4 とは異なり、DHCPv6 の IPv6 アドレス割り当てはメッセージ オプションを使用して処理されます。
- DHCP 検出や DHCP オファーなど、DHCPv4 でサポートされていたメッセージ タイプが DHCPv6 では削除されています。代わりに、DHCPv6 サーバはクライアントの送信要求メッセージとその後に続くサーバのアドバタイズ メッセージによって検索されます。
- DHCPv4 クライアントとは異なり、DHCPv6 クライアントは複数の IPv6 アドレスを要求できます。

DHCPv4 フェールオーバーによって、DHCP サーバのペアは、片方が機能を停止したらもう一方が引き継ぐという方法で機能します。このサーバのペアは、メイン サーバおよびバックアップ サーバと呼ばれます。通常の状況では、メイン サーバがすべての DHCP 機能を実行します。メイン サーバが使用不能になると、バックアップ サーバが引き継ぎます。このように、DHCP フェールオーバーでは、メイン サーバに障害が発生しても DHCP サービスへのアクセスが失われないようにします。

DNS

DNS サーバは、IP アドレス、ホスト名など、ネットワーク全体のホストに関する情報を格納します。DNS は主に IP アドレスとドメイン名を変換するためにこの情報を利用します。www.cisco.com などの名前を IP アドレスに変換することで、インターネットベースのアプリケーションへのアクセスが簡素化されます。

リース クエリー

リース クエリー機能を使用すると、プロビジョニング グループ内の Network Registrar DHCP サーバに、現在の IP アドレス情報を直接要求できます。デバイスの IP アドレスを見つけるために、RDU は DHCP リース クエリー メッセージを、デバイスのプロビジョニング グループ内の DHCP サーバだけに送信し、ネットワーク上のすべての DHCP サーバにクエリーが送信されないようにします。すべての応答の中で、デバイスと最後に通信したサーバからの応答が信頼できる応答と見なされます。

以前のバージョンの BAC では、リース クエリー機能は、リース クエリー要求を送信するための送信元インターフェイスと送信元ポートの選択をオペレーティング システムに依存していました。このリリースでは、特定のインターフェイスと送信元ポートを使用するように RDU を設定できます。

このリリースの BAC でのリース クエリー サポートの詳細については、P.6-19 の「リース クエリー」を参照してください。

Key Distribution Center

Key Distribution Center（KDC; 鍵発行局）は、PacketCable MTA を認証し、サービス チケットを MTA に与えます。そのため、MTA の証明書を検査するとともに、KDC 自体の証明書を提示して MTA が KDC を認証できるようにする必要があります。また、DPE（プロビジョニング サーバ）と通信して、MTA がネットワークでプロビジョニングされていることを検証します。

KDC は Solaris オペレーティング・システムを実行するサーバにインストールする必要があります。

KDC の認証に使用される証明書は BAC には同梱されていません。必要な証明書を Cable Television Laboratories, Inc.（CableLabs）から入手する必要があります。また、これらの証明書の内容は、MTA にインストールされているものと一致する必要があります。詳細については、P.14-3 の「PKCert.sh ツールの使用方法」を参照してください。

注意　証明書がインストールされていないと、KDC は機能しません。

KDC では、ライセンスが機能している必要があります。シスコ代理店から KDC ライセンスを入手し、正しいディレクトリにインストールしてください。ライセンスのインストール方法については、P.7-9 の「KDC ライセンス」を参照してください。

KDC には、BAC インストール中に *BPR_HOME/kdc/solaris/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、操作要件で指示された値に変更することができます。詳細については、P.7-7 の「KDC のデフォルト プロパティ」を参照してください。

また、KDC は複数のレルムの管理をサポートします。追加のレルム設定の詳細については、P.7-10 の「複数レルムのサポート」を参照してください。

BAC プロセス ウォッチドッグ

BAC プロセス ウォッチドッグは、すべての BAC プロセスのランタイム状況を監視する管理エージェントです。このウォッチドッグ プロセスにより、プロセスが予想外に停止した場合に自動的に再開されるようになります。BAC コンポーネントを実行する各システム上で、BAC プロセス ウォッチドッグのインスタンスが 1 つ実行されます。

BAC プロセス ウォッチドッグは、監視対象プロセスの状態を開始、停止、再開、決定するコマンドライン ツールとして利用できます。

監視対象プロセスの管理方法の詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP エージェント

BAC では、RDU サーバおよび DPE サーバについて基本的な SNMP v2 ベースの監視がサポートされます。BAC SNMP エージェントでは SNMP 通知と SNMP トラップがサポートされます。それらをまとめて「通知」と呼びます。

SNMP エージェントは、次の方法で設定できます。

- RDU で、SNMP 設定コマンドライン ツール（P.10-10 の「SNMP の使用によるサーバの監視」を参照）または API を介して設定する。
- DPE で、**snmp-server** CLI コマンドを使用して設定する。『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

管理者のユーザ インターフェイス

BAC 管理者のユーザ インターフェイスは、BAC システムを集中管理するための Web ベースのアプリケーションです。このシステムを使用して、次の作業を行うことができます。

- グローバル デフォルトの設定
- カスタム プロパティの定義
- サービス クラスの追加、修正、および削除
- DHCP 基準の追加、修正、および削除
- デバイスの追加、修正、および削除
- デバイスのグループ化
- サーバの状態とサーバ ログの表示
- ユーザの管理

このインターフェイスの使用方法については、それぞれ次の章を参照してください。

- 第 11 章「管理者のユーザ インターフェイスについて」：BAC 管理者のユーザ インターフェイスにアクセスする方法や設定方法について説明します。
- 第 12 章「管理者のユーザ インターフェイスの使用方法」：各種 BAC コンポーネントの監視など、管理作業を行う方法について説明します。
- 第 13 章「Broadband Access Center の設定」：BAC を設定するために実行する作業について説明します。

プロビジョニングの概念

この項では、プロビジョニングの重要な概念について説明します。次の項目を取り上げます。

- プロビジョニング グループ（P.2-16）
- 静的プロビジョニングと動的プロビジョニングの比較（P.2-16）
- プロビジョニング グループの機能（P.2-17）

プロビジョニング グループ

プロビジョニング グループは、通常 1 つ以上の DPE と DHCP サーバのフェールオーバー ペアで構成されるサーバを、論理的に（通常は地理的に）グループ化したものになるように設計されています。特定のプロビジョニング グループ内の各 DPE では、RDU からの同一の構成セットがキャッシュされます。その結果、冗長性とロード バランシングが可能になります。デバイスの数が増えたら、追加のプロビジョニング グループを配備に加えることができます。

（注）プロビジョニング　グループのサーバは、各地域に設置する必要はありません。中央のネットワーク オペレーション センターに簡単に配備できます。

プロビジョニング グループでは、BAC 配備のスケーラビリティを拡張する手段として、各プロビジョニング グループをデバイスのサブセットだけに関連付けます。このようなデバイスのパーティション化では、デバイスを地域的にグループ化することや、サービス プロバイダーによって定義されたポリシー別にグループ化することができます。

サービス プロバイダーで配備を拡大するには、次の作業を行います。

- 既存の DPE サーバのハードウェアをアップグレードする。
- プロビジョニング グループに DPE サーバを追加する。
- プロビジョニング グループを追加する。

冗長性とロード シェアリングをサポートするために、各プロビジョニング サポートは任意の数の DPE に対応できます。DHCP サーバから要求が届くと、その要求はプロビジョニング グループ内の DPE 間に分配され、デバイスと特定の DPE の間にアフィニティが確立されます。プロビジョニング グループ内の DPE 状態が安定している限り、このアフィニティは保たれます。

静的プロビジョニングと動的プロビジョニングの比較

BAC は、デバイス構成を使用してネットワークにデバイスをプロビジョニングします。デバイス構成とは、テクノロジー タイプに基づいた特定のデバイスのプロビジョニング データです。BAC を使用したデバイスのプロビジョニング方法には、静的プロビジョニングと動的プロビジョニングの 2 つがあります。

静的プロビジョニング時には、BAC システムに静的設定ファイルを入力します。この設定ファイルは、構成を生成するために TFTP を介して特定のデバイスに配信されます。BAC は、静的設定ファイルをその他のバイナリ ファイルと同様に扱います。

動的プロビジョニング時にはテンプレートを使用します。テンプレートは、DOCSIS、PacketCable、または CableHome オプションと、特定のサービス クラスで使用されると動的にファイルを生成する値が含まれるテキスト ファイルです。動的設定ファイルを使用すると、プロビジョニング プロセス中の柔軟性とセキュリティが強化されます。

表 2-4 に、それぞれのファイルを使用した静的プロビジョニングと動的プロビジョニングの影響を示します。

表 2-4　静的プロビジョニングと動的プロビジョニングの比較

静的ファイルを使用した 静的プロビジョニング	テンプレート ファイルを使用した 動的プロビジョニング
使用可能なサービス オファリングの数が少ない場合に使用する	使用可能なサービス オファリングの数が多い場合に使用する
限定的な柔軟性	より高い柔軟性（特にデバイスに固有の構成が必要な場合）
相対的に低い安全性	より高い安全性
高いパフォーマンス	低いパフォーマンス。これは、デバイスに割り当てられているテンプレートが更新されるたびに、そのテンプレートに関連付けられているすべてのデバイスの構成が更新されるためです。
簡単な使用方法	より複雑な使用方法

プロビジョニング グループの機能

配備にデバイスのサブセットをプロビジョニングするには、プロビジョニング グループによるそれらのデバイスのプロビジョニングが可能であり、かつイネーブルになっている必要があります。たとえば、DPE がこの機能をサポートするように設定されていない場合、プロビジョニング グループは PacketCable MTA を Secure モードでプロビジョニングできません。

以前のリリースの BAC では、プロビジョニング グループ内の各 DPE がサポートできる機能を起動時に RDU に登録していました。この情報が、プロビジョニング グループに含まれる他の DPE の情報と結合されて、グループがサポートできるデバイス タイプを決定していました。サーバは、その下位機能を登録し、またそれらの機能をイネーブルまたはディセーブルに登録しました。サーバの登録後、プロビジョニング グループは自動的にイネーブルになり、サポート可能なデバイス タイプをサポートしていました。一方、このリリースの BAC では、次のように手動でデバイス サポートをイネーブルにする必要があります。

- Provisioning Group Details ページの管理者のユーザ インターフェイスを使用します。P.12-29 の「プロビジョニング グループの表示」を参照してください。
- API で *ProvGroupCapabilitiesKeys* 定数を使用します。詳細については、API Javadoc を参照してください。

イベントのロギング

イベントのロギングは RDU と DPE で実行されます。まれに、視認性向上のために、DPE イベントが RDU に記録されることもあります。ログ ファイルはそれぞれのログ ディレクトリに保存され、任意のテキスト エディタを使用して調べることができます。ログ ファイルを圧縮すると、トラブルシューティングや障害の解決のために Cisco Technical Assistance Center またはシステム インテグレータに電子メールで送信しやすくなります。また、RDU と DPE のログには、管理者のユーザ インターフェイスからアクセスすることもできます。

ログのレベルと構造、およびログ ファイルの番号付けと循環の詳細については、P.10-2 の「ログのレベルおよび構造」を参照してください。

CHAPTER 3

設定のワークフロー

この章は 2 つの項で構成されており、各項でさまざまなテクノロジーをサポートするように Cisco Broadband Access Center（BAC）コンポーネントを設定する際のプロセスを定義します。次の項で構成されています。

（注）

Aapplication Programming Interface（API; アプリケーション プログラミング インターフェイス）を使用して、この章で説明する設定作業のすべてを実行することもできます。詳細については、4.0 API Javadoc を参照してください。

コンポーネントのワークフロー

この項では、BAC でサポートされるテクノロジーに合せて各 BAC コンポーネントを設定する際に必要なワークフローについて説明します。これらの設定作業を行ってから、特定のテクノロジーをサポートするように BAC を設定する必要があります。

BAC コンポーネントは、次の順序で設定する必要があります。

1. RDU ワークフロー（P.3-2）
2. DPE ワークフロー（P.3-2）
3. Network Registrar ワークフロー（P.3-4）

RDU ワークフロー

表 3-1 は、RDU 設定時のワークフローを示しています。

表 3-1　RDU の設定ワークフロー

	タスク	参照先
ステップ 1	BAC に利用されるシステム syslog サービスを設定する。	*Installation and Setup Guide for Cisco Broadband Access Center 4.0*
ステップ 2	BAC 管理者のユーザ インターフェイスにアクセスする。	管理者のユーザ インターフェイスへのアクセス（P.11-3）
ステップ 3	ログイン パスワードを変更する。	管理者のユーザ インターフェイスへのアクセス（P.11-3）
ステップ 4	ライセンス ファイルを追加する。	ライセンスの管理（P.13-23）
ステップ 5	RDU データベースをバックアップする。	バックアップと回復（P.15-5）
ステップ 6	RDU SNMP エージェントを設定する。	snmpAgentCfgUtil.sh ツールの使用方法（P.10-11）
ステップ 7	デフォルトの重大度ログ レベル（Notification レベル）を設定する。	RDU ログ レベル ツールの使用方法（P.10-5）
ステップ 8	IPv4 または IPv6 のプロビジョニング グループ機能をイネーブルにする。	プロビジョニング グループの表示（P.12-29）

DPE ワークフロー

このワークフローに示されている作業は、表 3-1 に示されている作業の後で実行する必要があります。次のプロトコルをサポートするように DPE を設定できます。

- IPv4。表 3-2 を参照してください。
- IPv6。表 3-3 を参照してください。

（注） アスタリスク（*）が付いている作業は必須です。

表 3-2 は、DPE を IPv4 用に設定するワークフローを示しています。

表 3-2　IPv4 用 DPE 設定ワークフロー

	タスク	参照先
ステップ 1	BAC に利用されるシステム syslog サービスを設定する。	『*Installation and Setup Guide for Cisco Broadband Access Center 4.0*』
ステップ 2	パスワードを変更する。	**password** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 3	プロビジョニング インターフェイスを設定する。*	**interface ip** *ipv4_address* **provisioning** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 4	プロビジョニング FQDN を設定する。	**interface ip** *ipv4_address* **provisioning fqdn** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 5	Cisco Network Registrar 拡張と通信するインターフェイスを設定する。	**interface ip** *ipv4_address* **pg-communication** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 6	BAC の共有秘密情報を設定する。*	**dpe shared-secret** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 7	RDU に接続するために DPE を設定する。*	**dpe rdu-server port** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 8	Network Time Protocol（NTP; ネットワーク タイム プロトコル）を設定する。	Solaris のマニュアルに示されている設定情報
ステップ 9	プライマリ プロビジョニング グループを設定する。*	**dpe provisioning-group primary** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 10	DPE SNMP エージェントを設定する。	SNMP エージェント コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	（注） SNMP エージェントを設定するには、DPE のコマンドライン インターフェイスまたは **snmpAgentCfgUtil.sh** ツールを使用します（P.10-11 の「snmpAgentCfgUtil.sh ツールの使用方法」を参照）。	
ステップ 11	RDU に接続されていることを確認する。	サーバの表示（P.12-23）
ステップ 12	v4 のプロビジョニング グループ機能をイネーブルにする。	プロビジョニング グループの表示（P.12-29）

表 3-3 は、DPE を IPv6 用に設定するワークフローを示しています。ここでは、IPv6 に関する作業だけを説明します。DPE の基本的な設定を行うには、表 3-2 の作業を完了してから、この表に示す手順を追加で実行してください。

表 3-3　IPv6 用 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 インターフェイスを設定する。*	**interface ip** *ipv6_address* **provisioning** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 2	プロビジョニング FQDN を設定する。	**interface ip** *ipv6_address* **provisioning fqdn** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 3	TFTP をイネーブルにする。	**service tftp** *1..1* **ipv6 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 4	ToD をイネーブルにする。	**service tod** *1..1* **ipv6 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 5	DPE をリロードする。	**dpe reload** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 6	v6 のプロビジョニング グループ機能をイネーブルにする。	プロビジョニング グループの表示（P.12-29）

Network Registrar ワークフロー

このワークフローに示されているアクティビティは、表 3-2 に示されている作業の後で実行する必要があります。

注意　BAC DHCP オプション設定は、Cisco Network Registrar 内で設定された DHCP オプション値よりも常に優先されます。

Network Registrar を設定するには、次の手順に従います。

- DHCPv4 の場合は、表 3-4 を参照してください。
- DHCPv6 の場合は、表 3-5 を参照してください。

（注）　アスタリスク（*）が付いている作業は必須です。

表 3-4 は、Network Registrar を DHCPv4 用に設定するワークフローを示しています。

表 3-4　DHCPv4 の Network Registrar ワークフロー

	タスク	参照先
ステップ 1	Network Registrar 拡張を検証する。	*Installation and Setup Guide for Cisco Broadband Access Center 4.0*
ステップ 2	BAC に利用されるシステム syslog サービスを設定する。	*Installation and Setup Guide for Cisco Broadband Access Center 4.0*

表 3-4　DHCPv4 の Network Registrar ワークフロー（続き）

	タスク	参照先
ステップ 3	RDU に定義されているものと一致するクライアント クラス / 選択タグを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 4	ポリシーを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 5	スコープを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 6	Network Registrar データベースをバックアップする。	*User Guide for Cisco Network Registrar 7.0*
ステップ 7	正しい RDU に接続されていることを確認する。	サーバの表示（P.12-23）
ステップ 8	DHCP サーバをリロードする。	*User Guide for Cisco Network Registrar 7.0*

表 3-5 は、Network Registrar を DHCPv6 用に設定するワークフローを示しています。DOCSIS ケーブル モデム、コンピュータ、PacketCable MTA を含む、プロビジョニングされているデバイスおよびプロビジョニングされていないデバイスのカテゴリごとにこの作業リストを実行します。

表 3-5　DHCPv6 の Network Registrar ワークフロー

	タスク	参照先
ステップ 1	Network Registrar 拡張を検証する。	*Installation and Setup Guide for Cisco Broadband Access Center 4.0*
ステップ 2	BAC に利用されるシステム syslog サービスを設定する。	*Installation and Setup Guide for Cisco Broadband Access Center 4.0*
ステップ 3	RDU に定義されているものと一致するクライアント クラス / 選択タグを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 4	ポリシーを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 5	リンクを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 6	プレフィックスを設定する。プレフィックスごとに、必ず適切なポリシー、リンク、および選択タグを設定します。* **（注）** ケーブル モデムなど、一部の DHCP クライアントは、複数の IPv6 アドレスが含まれるオファーを拒否します。プレフィックスを定義するときには、1 つのクライアントに複数の IPv6 アドレスを割り当てないように Network Registrar を設定します。2 つのプレフィックスに同じ選択タグを追加しないでください。追加すると、Network Registrar が各プレフィックスから 1 つずつ IP アドレスを選択して、2 つの IP アドレスをクライアントに割り当ててしまうためです。	*User Guide for Cisco Network Registrar 7.0*
ステップ 7	Network Registrar データベースをバックアップする。	*User Guide for Cisco Network Registrar 7.0*
ステップ 8	正しい RDU に接続されていることを確認する。	サーバの表示（P.12-23）
ステップ 9	DHCP サーバをリロードする。	*User Guide for Cisco Network Registrar 7.0*

技術のワークフロー

この項では、特定の技術をサポートするように BAC を設定する際に必要な作業について説明します。次の項目を取り上げます。

- DOCSIS ワークフロー（P.3-6）
- PacketCable ワークフロー
  - PacketCable Secure（P.3-7）
  - PacketCable Basic（P.3-9）
- CableHome ワークフロー（P.3-11）

（注） アスタリスク（*）が付いている作業は必須です。

DOCSIS ワークフロー

BAC は、DOCSIS 仕様のバージョン 1.0、1.1、2.0、3.0 をサポートしています。

DOCSIS を操作するために BAC を正常に設定するには、この項で説明する作業に加えて、P.3-2 の「コンポーネントのワークフロー」で示す作業も実行する必要があります。

表 3-6 は、DOCSIS をサポートするように BAC を設定するワークフローを示しています。

表 3-6　DOCSIS ワークフロー

	タスク	参照先
ステップ 1	RDU を設定する。	
	a. プロビジョニングされる DHCP 基準をすべて設定する。	DHCP 基準の設定（P.13-15）
	b. プロビジョニングされるサービス クラスを設定する。	サービス クラスの設定（P.13-2）
	c. 無差別モードの操作を設定する。	システム デフォルト（P.13-12）
ステップ 2	DPE を設定する。	
	a. TFTP サービスをイネーブルにする。	**service tftp** *1..1* **ipv4 | ipv6 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	b. オプションで、ToD サービスをイネーブルにする。	**service tod** *1..1* **ipv4 | ipv6 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 3	Network Registrar を設定する。	
	クライアント クラス / 選択タグを、プロビジョニングされる DOCSIS モデムの DHCP 基準に追加されたものと一致するように設定する。	*User Guide for Cisco Network Registrar 7.0*

PacketCable ワークフロー

BAC は、PacketCable 仕様のバージョン 1.0、1.1、1.5 をサポートしています。

また BAC は、PacketCable 音声サービスの 2 つのバリアント（デフォルトの Secure モードとノンセキュア Basic モード）もサポートしています。PacketCable Basic は、ノンセキュア バリアントでセキュリティが制限されている点を除けば、標準の PacketCable とほとんど同じです。

この項では、バリアントごとに実行する必要がある作業を示します。

- PacketCable Secure（P.3-7）
- PacketCable Basic（P.3-9）

（注） この項のワークフローは、適切な PacketCable 設定ファイルと正しい MIB が読み込まれていることを前提にしています。

PacketCable Secure

BAC は PacketCable Secure の次の 2 つのバリアントをサポートしています。

- 北米版 PacketCable
- 欧州版 PacketCable

Euro-PacketCable サービスは、北米版 PacketCable 仕様に相当する欧州版です。両者の唯一の大きな違いは、Euro PacketCable では異なる MIB が使用される点です。詳細については、P.7-33 の「Euro-PacketCable MIB」を参照してください。

この項で示す PacketCable 関連の作業は、P.3-2 の「コンポーネントのワークフロー」に示されている作業の後で実行する必要があります。

（注） PacketCable に準拠した操作を行う場合、MTA、KDC、DPE 間の最大許容クロック スキューは 300 秒（5 分）です。この値はデフォルト設定です。

表 3-7 は、PacketCable Secure をサポートするように BAC を設定するワークフローを示しています。

（注） アスタリスク（*）が付いている作業は必須です。

表 3-7　PacketCable Secure ワークフロー

	タスク	参照先
ステップ 1	RDU を設定する。	
	a. Multimedia Terminal Adapter（MTA; マルチメディア ターミナル アダプタ）FQDN の自動ネゴシエーションをイネーブルにする。	自動 FQDN 生成（P.13-32）
	b. プロビジョニングされる DHCP 基準をすべて設定する。	DHCP 基準の設定（P.13-15）

表 3-7　PacketCable Secure ワークフロー（続き）

	タスク	参照先
	c. プロビジョニングされるサービス クラスをすべて設定する。	サービス クラスの設定（P.13-2）
	d. SNMPv3 クローニング キーを設定する。*	PacketCable MTA と安全に通信するための RDU と DPE 上での SNMPv3 クローニングの設定（P.7-30）
	e. Euro PacketCable を使用している場合、Euro-PacketCable の MIB を使用するように RDU を設定する。	Euro-PacketCable MIB の設定（P.7-34）
ステップ 2	DPE を設定する。	
	a. KDC サービス キーを設定する。*	**service packetcable** *1..1* **registration kdc-service-key** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	b. プライバシー ポリシーを設定する。*	**service packetcable** *1..1* **registration policy-privacy** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	c. SNMPv3 クローニング キーを設定する。*	**service packetcable** *1..1* **snmp key-material** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	d. PacketCable をイネーブルにする。*	**service packetcable** *1..1* **enable** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	e. TFTP サービスをイネーブルにする。	**service tftp ipv4 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	f. オプションで、ToD サービスをイネーブルにする。	**service tod ipv4 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	g. オプションで、MTA ファイル暗号化を設定する。	**service packetcable** *1..1* **registration encryption enable** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 3	KDC を設定する。	
	a. シスコ代理店から KDC ライセンスを入手する。	KDC ライセンス（P.7-9）
	b. PKCert.sh ツールを使用して、証明書チェーンを設定する。Euro PacketCable の場合は、**-e** オプションを使用します。	PKCert.sh ツールの使用方法（P.14-3）
	c. DPE のプロビジョニング FQDN ごとにサービス キーペアを設定する。	KeyGen ツールの使用方法（P.14-9）
	d. Ticket-granting-ticket（TGT; チケット認可チケット）のサービス キーを設定する。	KeyGen ツールの使用方法（P.14-9）
	e. ネットワーク タイム プロトコル（NTP）を設定する。	Solaris の NTP 設定に関する詳細については Solaris のマニュアルを参照
ステップ 4	DHCP を設定する。	
	a. 必要なすべての PacketCable プロパティを設定する。	KeyGen ツールの使用方法（P.14-9）
	b. MTA スコープの動的 DNS を設定する。	*User Guide for Cisco Network Registrar 7.0*

表 3-7　PacketCable Secure ワークフロー（続き）

	タスク	参照先
	c. クライアント クラス/スコープ選択タグを、プロビジョニングされる PacketCable MTA の DHCP 基準に追加されたものと一致するように設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 5	DNS を設定する。	
	a. DHCP サーバごとに動的 DNS を設定する。	*User Guide for Cisco Network Registrar 7.0*
	b. KDC レルムのゾーンを設定する。	*User Guide for Cisco Network Registrar 7.0*

PacketCable Basic

この項で示す PacketCable 関連の作業は、P.3-2 の「コンポーネントのワークフロー」に示されている作業の後で実行する必要があります。

表 3-8 は、PacketCable Basic を BAC に設定するワークフローを示しています。

（注） アスタリスク（*）が付いている作業は必須です。

表 3-8　PacketCable Basic ワークフロー

	タスク	参照先
ステップ 1	DPE を設定する。	
	a. PacketCable をイネーブルにする。*	**service packetcable** *1..1* **enable** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	b. TFTP サービスをイネーブルにする。	**service tftp** *1..1* **ipv4 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	c. オプションで、ToD サービスをイネーブルにする。	**service tod** *1..1* **ipv4 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 2	DHCP を設定する。	
	a. MTA スコープの動的 DNS を設定する。	*User Guide for Cisco Network Registrar 7.0*
	b. プロビジョニングされる PacketCable MTA の DHCP 基準に追加されたものと一致するクライアント クラス / スコープ選択タグを設定する。*	*User Guide for Cisco Network Registrar 7.0*
ステップ 3	DNS を設定する。	
	DHCP サーバごとに動的 DNS を設定する。	*User Guide for Cisco Network Registrar 7.0*

表 3-8　PacketCable Basic ワークフロー（続き）

	タスク	参照先
ステップ 4	サービス クラスを設定する。このサービス クラスには次のプロパティが含まれている必要があります。	
	a. */pktcbl/prov/flow/mode* このプロパティは、MTA が使用する特定のフローを指定します。このプロパティを次のいずれかに設定します。 － BASIC.1：BASIC.1 フローを実行する。 － BASIC.2：BASIC.2 フローを実行する。 **（注）** このプロパティは、デバイス プロパティ階層の任意の場所で設定できます。	サービス クラスの設定（P.13-2）
	b. */cos/packetCableMTA/file* このプロパティには、MTA に提示される設定ファイルの名前が含まれます。この設定ファイルは、ファイルとして BAC に保存されます。 Basic MTA に提示される設定ファイルには、Basic の完全性ハッシュが含まれている必要があります。動的設定テンプレートを使用している場合、ハッシュはテンプレート処理中に透過的に挿入されます。動的テンプレートは、Secure モードと Basic モードの両方のプロビジョニングに使用できます。 ただし、Secure と Basic の静的設定ファイルは相互運用できないため、ファイルが Secure 静的設定ファイルの場合は Basic 静的設定ファイルに変換する必要があります。この変換の実行の詳細については、P.5-37 の「PacketCable Basic フローの有効化」を参照してください。	サービス クラスの設定（P.13-2）

CableHome ワークフロー

ノンセキュア CableHome テクノロジーを使用してプロビジョニング用に BAC を正常に設定するには、この項で説明する作業に加えて、P.3-2 の「コンポーネントのワークフロー」で示す作業も実行する必要があります。

表 3-9 に、CableHome をサポートするために BAC に対して実行する必要がある作業を示します。

表 3-9　CableHome ワークフロー

	タスク	参照先
ステップ 1	RDU を設定する。	
	a. プロビジョニングされる DHCP 基準を設定する。 プロビジョニングされるノンセキュア CableHome デバイスが使用する DHCP 基準をすべて追加します。	DHCP 基準の設定（P.13-15）
	b. プロビジョニングされるサービス　クラスを設定する。 プロビジョニングされるノンセキュア CableHome デバイスが使用する可能性があるサービス クラスを追加します。	サービス クラスの設定（P.13-2）
	c. 無差別モードの操作を設定する。	システム デフォルト（P.13-12）
ステップ 2	DPE を設定する。	
	a. TFTP サービスをイネーブルにする。	**service tftp** *1..1* **ipv4 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
	b. オプションで、ToD サービスをイネーブルにする。	**service tod** *1..1* **ipv4 enabled true** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）
ステップ 3	Network Registrar を設定する。	
	クライアント クラス / スコープ選択タグを、プロビジョニングされるノンセキュア CableHome の DHCP 基準に追加されたものと一致するように設定する。	*User Guide for Cisco Network Registrar 7.0*

CHAPTER 4

CPE プロビジョニングの概要

この章では、Broadband Access Center（BAC）で Customer Premises Equipment（CPE; 顧客宅内装置）がサポートするテクノロジーを使用して CPE を管理する方法について説明します。この章は、次の項で構成されています。

概要

BAC を使用して、住宅用デバイス（DOCSIS ケーブル モデムおよびセットトップ ボックス、PacketCable eMTA、CableHome デバイス、コンピュータ）のプロビジョニングと管理を行うことができます。

BAC は次のデバイス タイプのプロビジョニングを行います。

- DOCSIS 1.0、1.1、2.0 準拠のケーブル モデムと STB
- PacketCable バージョン 1.x 準拠の embedded Multimedia Terminal Adapter（eMTA; 組み込み型マルチメディア ターミナル アダプタ）
- CableHome 1.0 準拠のデバイス
- コンピュータ

このリリースの BAC は、次のプロビジョニングと管理をサポートします。

- IPv6 デバイス。次のものがあります。
  - DOCSIS 3.0 準拠のケーブル モデム
  - コンピュータ
  - Set-top box（STB; セットトップ ボックス）
- ビデオ STB(特に進化途上の OpenCable アプリケーション プラットフォームに基づく RNG-200 STB)。
- embedded Service/Application Functional Entities（eSAFE）デバイスのバリアント。混在 IP モードの PacketCable Multimedia Terminal Adapters（MTA; マルチメディア ターミナル アダプタ）などがあります。混在 IP モード MTA は、IPv6 組み込みケーブル モデムと IPv4 eMTA で構成される eSAFE デバイスです。このクラスのデバイスでは、ケーブル モデムにパケット テレフォニー、ホーム ネットワーキング、動画などの付加的機能が組み込まれています。

デバイス オブジェクト モデル

BAC のデバイス オブジェクト モデルは、DPE のデバイス管理用に生成される構成を制御する上で重要です。このデバイス構成生成プロセスは RDU で発生し、名前付き属性および関連付けを通じて制御されます。

デバイス オブジェクト モデルには、次の主要オブジェクトがあります。

- IP デバイス：プロビジョニングを必要とするネットワーク エンティティを表します。
- オーナー ID：加入者の外部識別子を表します。
- デバイス タイプ：デバイスのタイプを表します。
- プロビジョニング グループ：特定の複数の DPE からサービスを受けるデバイスの論理グループを表します。
- サービス クラス：デバイスに割り当てる構成プロファイルを表します。
- DHCP 基準：デバイスが、Cisco Network Registrar DHCP サーバ内で IP アドレスの選択を決定するための基準を表します。
- ファイル：プロビジョニングで使用される、テンプレートを含むファイルのコンテナとして機能します。
- ノード：デバイスをグループ化するための顧客固有のメカニズム。

BAC デバイス データ モデルの各種オブジェクトには、次の共通要素があります。

- 名前：たとえば、Gold サービス クラス。
- 属性：たとえば、Device ID や Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）。
- 関連付け：たとえば、デバイスとサービス クラスの関連付け。
- プロパティ：たとえば、デバイスを特定のプロビジョニング グループに追加することを指定するプロパティ。

図 4-1 に、デバイス データ モデルの各種オブジェクト間のインタラクションを示します。

図 4-1　デバイス オブジェクト モデル

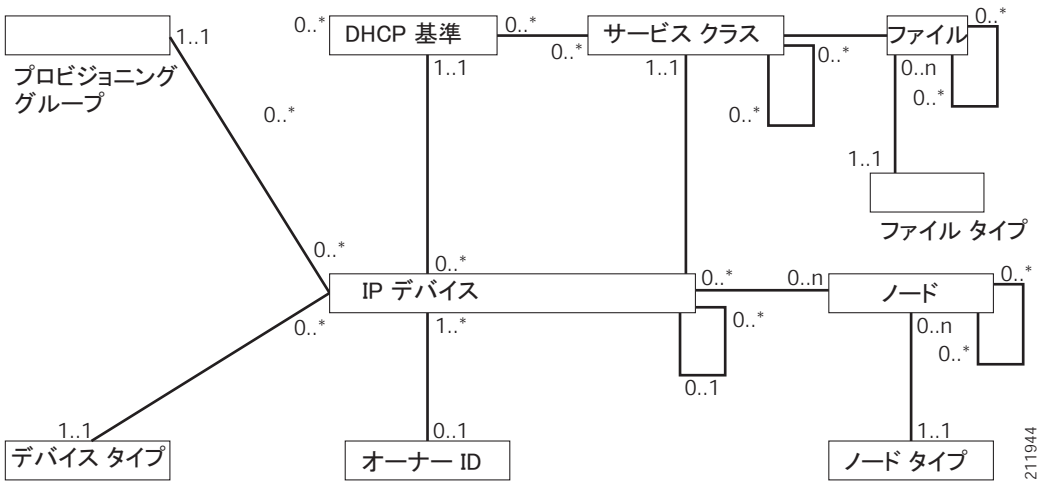


表 4-1 は、データ モデルの各オブジェクトに固有の属性および関連付けを示しています。

表 4-1　デバイス オブジェクトの関連付け

オブジェクト	関連付けられるオブジェクト
IP デバイス • 事前プロビジョニングまたはセルフプロビジョニングが可能です（P.4-9 の「BAC におけるデバイス配備」を参照）。 • 属性には、Device ID（MAC アドレスまたは DUID）や FQDN があります。	• オーナー ID • プロビジョニング グループ • サービス クラス • DHCP 基準 • デバイス タイプ
オーナー ID • デバイスに関連付けられます。そのため、デバイスに関連付けられた場合に限り存在します。 • グループ化をイネーブルにします。たとえば、*Joe* に属しているデバイスすべてをグループ化できます。	IP デバイス
デバイス タイプ • ある技術を持つデバイスすべてに共通したデフォルトが格納されます。 • グループ化をイネーブルにします。たとえば、すべての PacketCable デバイスをグループ化できます。	IP デバイス
ファイル プロビジョニングで使用されるファイルが格納されます（たとえば、設定ファイルやテンプレート）。	サービス クラス
サービス クラス 属性には、Type、Name、および Properties があります（詳細については、P.4-3 の「サービス クラス」を参照してください）。	• IP デバイス • ファイル • DHCP 基準 • 設定テンプレート（オプション）
DHCP 基準 グループ化をイネーブルにします。たとえば、特定の技術内のデバイスを別々のクラスの IP にグループ化できます。	• IP デバイス • サービス クラス • 設定テンプレート（オプション）

サービス クラス

サービス クラスは RDU 抽象の 1 つで、静的ファイルまたはテンプレート ファイルとしてデバイスに渡されるファイル構成を表します。サービス クラスを使用すると、デバイスをグループ化して構成セットにすることができます。構成セットは、各種のサービス レベルまたはパッケージで CPE に提供されます。

サービス クラスには次の種類があります。

- 登録されているもの：デバイスの登録時にユーザによって指定されます。このサービス クラスは、明示的に、Application Programming Interface（API; アプリケーション プログラミング インターフェイス）を介してデバイス レコードに追加されます。
- 選択されているもの：RDU 拡張によって選択され、返されます。
- 関連付けられているもの：登録、選択、またはその両方の操作によってデバイスに関連付けられます。このサービス クラスは、RDU 拡張によって選択されます。

デバイス用の選択されているサービス クラスが変更された場合は、デバイス構成が再生成されます。デバイス用の登録されているサービス クラスが変更された場合、それが選択されているサービス クラスでなくても、デバイス構成が再生成されます。その理由は、サービス クラスが適用するポリシーによって、選択されているサービス クラスが変更される可能性があるためです。

検出されたデータ

プロビジョニング プロセスの間、BAC は一連のプロパティを使用してデバイス タイプを検出し（デバイスがケーブル モデムやコンピュータかどうかなど）、そのデバイス タイプと技術用の構成を生成します。この一連のプロパティを使用して BAC が検出した情報は、検出されたデータと呼ばれます。BAC は、デバイスごとに検出されたデータを RDU データベースに格納します。

デバイスがプロビジョニング サーバに接続すると、ファームウェアのバージョン、MAC アドレス、操作モードなど、そのデバイス自体に関する詳細情報を提供します。プロビジョニング サーバに接続するケーブル モデムの場合、これらの詳細は次のメッセージに含まれています。

- IPv4 デバイスの検出メッセージ
- IPv6 デバイスの送信要求メッセージ

Network Registrar にインストールされている BAC 拡張も、デバイスの構成を要求するときに、検出されたデータを取得して RDU に送信します。これらのデバイスの場合、検出されたデータは Network Registrar の設定によって異なります。属性またはオプションが Network Registrar で使用できるように設定されている場合、拡張は、その属性またはオプションの値を DHCP パケットから取得し、検出されたデータの中に含めて BAC プロビジョニングで使用できるようにします。

表 4-2 は、BAC が IPv4 デバイスから検出するデータの一覧を示します。

表 4-2　IPv4 デバイスから検出されるデータ

オプション	説明
chaddr	クライアントのハードウェア アドレスを指定します。
client-id	クライアントで定義され、クライアントを一意に識別するバイト シーケンスまたは文字列を示します。
client-id-created-from-mac-address	クライアントの MAC アドレスから作成されたクライアントの識別子を示します。
dhcp-message-type	DHCP 検出や DHCP 確認など、DHCP メッセージのタイプを指定します。
giaddr	DHCP サーバの応答先となる IP アドレスを指定します。
hlen	ハードウェア アドレスの長さを指定します。
htype	ハードウェア タイプを指定します。
relay-agent-circuit-id	クライアントとサーバ間の DHCP パケットの送信元となる回線のエージェント ローカル識別子を符号化します。
relay-agent-info	CableLabs Relay Agent CMTS Capabilities オプションへのアクセスに使用します。
relay-agent-remote-id	回線のリモート ホスト側に関する情報を符号化します。
v-i-vendor-opts	クライアントからサーバに要求されたオプションを示します。
vendor-encapsulated-options	カプセル化されて標準の DHCP オプションに送信されるオプションを定義します。
vendor-class	DHCPv4 クライアントと関連付けられた CPE の機能を示す文字列が含まれます。

表 4-3 は、BAC が IPv6 デバイスから検出するデータの一覧を示します。

表 4-3　IPv6 デバイスから検出されるデータ

オプション	説明
peer-address	最初にメッセージを送信したクライアント、またはメッセージをリレーした直前のリレー エージェントの IPv6 アドレスを指定します。
link-address	クライアントのサブネットに接続するインターフェイスに割り当てられている非リンクローカル アドレスを指定します。
client-identifier	リース用のクライアントの DHCP Unique Identifier（DUID）を指定します。DHCPv6 クライアントはクライアントのハードウェア アドレス（chaddr）を使用できないため、IPv6 環境でデバイスを一意に識別するには DUID を使用します。この情報は、DHCP 送信要求メッセージから入手できます。
oro	要求されたオプションを示します。
vendor-opts	ベンダー固有の情報を交換するためにクライアントとサーバで使用されるベンダー固有の情報オプションを示します。この情報は、DHCP 送信要求メッセージから入手できます。
vendor-class	クライアントが実行されているハードウェアを製造したベンダーを示します。この情報は DHCPv6 送信要求メッセージから入手できます。

Device Details ページの管理者のユーザ インターフェイスを使用して検出されたデータを表示できます。デバイスの詳細の表示方法については、P.12-10 の「デバイスの詳細の表示」を参照してください。

BAC 拡張が DHCPv4 および DHCPv6 のデータの検出に使用するプロパティの一覧については、P.6-15 の「属性とオプション」を参照してください。

DUID と MAC アドレスの比較

DHCPv4 標準は、クライアント識別子または MAC アドレスを DHCP クライアントのプライマリ デバイス識別子として使用します DHCPv6 では、新しいプライマリ デバイス識別子である DHCP Unique Identifier（DUID）が導入されています。

DHCPv4 は、アドレスの割り当て時に、ハードウェア アドレスとオプションのクライアント識別子を使用してクライアントを識別します。DHCPv6 も基本的に同じスキームに従いますが、クライアント識別子が必須になり、ハードウェア アドレスとクライアント ID を連結して 1 つの一意のクライアント識別子にします。

DHCPv6 のクライアント識別子は次の項目で構成されます。

- DUID：クライアント システムを示します（DHCPv4 のようにインターフェイスだけではない）。
- Identity Association Identifier（IAID）：システム上のインターフェイスを示します。RFC 3315 で説明されているように、Identity Association は、サーバとクライアントが、関連する一連の IPv6 アドレスを識別、グループ化、および管理するために使用する手段です。

DHCP クライアントとサーバはそれぞれ DUID を持っています。DHCP サーバは、DUID を使用してクライアントを IA と関連付けて識別し、設定情報を選択します。DHCP クライアントは、サーバの識別が必要なメッセージの場合、DUID を使用してメッセージ内のサーバを識別します。

構成の生成と処理

BAC 配備内でデバイスを有効にすると、そのデバイスは BAC サーバとの交信を開始します。交信が確立すると、デバイスの事前設定されたポリシーが、デバイスに関連付けられている設定テンプレートに基づいて DPE によるデバイスのプロビジョニングおよび管理を決定します。このデバイスの信頼できるプロビジョニング情報は、デバイス構成として、RDU から DPE に転送されます。DPE はデバイス構成をキャッシュし、デバイスからのサービス要求に対して使用します。

デバイス構成には、ユーザが必要とする次のプロビジョニング情報を含めることができます。

- DHCP IP アドレス選択
- 帯域幅
- データ レート
- フロー制御
- 通信速度
- サービス レベル（サービス クラスとも呼ばれる）

構成には、識別子（MAC アドレスまたはファイル名）と、構成が再生成されるたびに大きくなるリビジョン番号が含まれます。

RDU は、次の場合にデバイスの構成を再生成します。

- デバイスのサービス クラス変更など、特定のプロビジョニング API コールが実行された場合。
- 構成の検証が失敗した場合。デバイスから送信された DHCP 要求の特定のパラメータが最初の要求パラメータと異なる場合などに発生します。

RDU がデバイスの構成を再生成するたびに、更新された構成が該当する DPE に転送され、キャッシュされます。

この項では、次の関連する概念についても説明します。

- 静的ファイルとテンプレート ファイルの比較（P.4-6）
- プロパティ階層（P.4-7）
- テンプレートとプロパティ階層（P.4-7）
- カスタム プロパティ（P.4-8）

静的ファイルとテンプレート ファイルの比較

BAC でのデバイスのプロビジョニングは、静的ファイルとテンプレート ファイルという 2 つのタイプの設定ファイルを使用して行うことができます。

静的設定ファイルを使用する場合、ファイルを BAC システムに入力します。設定ファイルが入力されたら、その構成を生成するために TFTP を介して特定のデバイスに配信されます。BAC は、静的設定ファイルをその他のバイナリ ファイルと同様に扱います。静的ファイルは、拡張子 *.cm* で識別します。

テンプレート ファイルは、DOCSIS、PacketCable、または CableHome オプションと、特定のサービス クラスで使用された場合に動的ファイル生成を実行する値が含まれるテキスト ファイルです。BAC には設定ファイル ユーティリティが同梱されており、DOCSIS、PacketCable、および CableHome の設定ファイルとテンプレート ファイルのテスト、検証、表示に使用できます。設定ファイルユーティリティの使用方法の詳細については、P.5-23 の「設定ファイル ユーティリティの使用方法」を参照してください。テンプレート ファイルは、拡張子 *.tmpl* で識別します。

静的プロビジョニングと動的プロビジョニングの概要については、表 2-4 を参照してください。

プロパティ階層

BAC のプロパティを使用すると、BAC で API を介してデータにアクセスしたり、データを格納したりできます。事前プロビジョニングされたデータ、検出されたデータ、およびステータス データは、API を介して、対応するオブジェクトのプロパティから取得できます。また、プロパティを使用すると、BAC を適切な粒度で（システム レベルからデバイス グループおよび個々のデバイスのレベルまで）設定できます。

デバイス関連のプロパティは、BAC プロパティ階層内の任意の許容ポイントで定義できます。プロパティを任意のレベルで割り当てられるかどうかの詳細については、API Javadoc を参照してください。

BAC プロパティ階層には、個々のデバイスまたはデバイスのグループに対してプロパティを定義できるという柔軟性があります。プロパティは、デバイスおよび関連付けられたオブジェクト上で見つかるまで次の順序で検索されます。

1. デバイス登録済みプロパティ：API または管理者のユーザ インターフェイスを介して設定されたプロパティを指定します。
2. デバイス選択プロパティ：サービス レベル選択プロセスによってデバイス レコードに保存されたプロパティを指定します。
3. デバイス検出プロパティ：デバイス検出プロセスによってデバイス レコードに保存されたプロパティを指定します。
4. プロビジョニング グループ：デバイスのプロビジョニング グループのプロパティを指定します。
5. サービス クラス：デバイスのサービス クラス上に設定されているプロパティを指定します。サービス レベル選択プロセスがデバイスの選択済みサービス クラスを判別したら、そのオブジェクトのプロパティが使用されます。判別しなかった場合、プロパティは、API または管理者のユーザ インターフェイスを介して、デバイスに対して設定されている登録済みサービス クラスから検索されます。
6. DHCP 基準：デバイスの DHCP 基準上に設定されているプロパティを指定します。サービス レベル選択プロセスがデバイスの選択済み DHCP 基準を判別したら、そのオブジェクトのプロパティが使用されます。判別しなかった場合、プロパティは、API または管理者のユーザ インターフェイスを介して、デバイスに対して設定されている登録済み DHCP 基準から検索されます。
7. テクノロジー デフォルト：デバイスのテクノロジー デフォルトに設定されているプロパティを指定します。たとえば、DOCSIS モデム、PacketCable MTA、またはコンピュータのテクノロジー デフォルトなどです。
8. システム デフォルト：システム デフォルトに設定されているプロパティを指定します。

テンプレートとプロパティ階層

動的に構成を生成すると、デバイス設定ファイルのテキスト記述（テンプレートとも呼ばれる）をバイナリのデバイス構成に変換する処理も行われます。このバイナリ設定ファイルは、基本的には type-length-value（TLV）タプルのリストであり、各タプルにはデバイス構成の設定値が含まれます。処理後のバイナリ構成は、TFTP を介してデバイスに転送されます。

動的な構成生成では、マクロ機能の使用によって大幅な柔軟性が実現されます。マクロにより、BAC プロパティ階層の値をテンプレートに代入できます。この代入は、通常は上書きされる次のような値に使用されます。

- ダウンストリームまたはアップストリームの帯域幅
- ケーブル モデムの背後にあるデバイスの数

このように、BAC は 1 つのテンプレートを使用して、数台から数百や数千、数百万台まで、何台でもデバイスの構成を生成します。

カスタム プロパティ

BAC を使用すると、RDU 内に新しいプロパティを定義できます。定義されたプロパティは、API を介して任意のオブジェクト上に格納することができます。これらのプロパティによって、テンプレートに値を代入できます。

カスタム プロパティは、RDU で定義された変数名であるため、スペースを含めることはできません。

カスタム プロパティの作成方法の詳細については、P.13-6 の「カスタム プロパティの設定」を参照してください。

BAC におけるデバイス配備

BAC 配備はプロビジョニング グループに分割され、各プロビジョニング グループはデバイスのサブセットだけに関連付けられます。耐障害性を確保するため、プロビジョニング グループによって提供されるサービスはすべて実装されます（P.2-16 の「プロビジョニング グループ」を参照）。

BAC では、次の 2 つのデバイス配備オプションを使用できます。

- Preprovisioned：RDU に、さまざまなデバイス タイプの構成とルールが読み込まれます。デバイス レコードが RDU に追加されると、そのデバイス タイプ固有の構成がマッピングされます。
- Self-provisioned：デバイスがプロビジョニング グループと最初に交信した後で、デバイス レコードが RDU に追加されます。ただし、事前プロビジョニングされたルールによってデバイスの構成が決まります。

この項では、次のトピックについて取り上げます。

CPE 登録モード

登録モードによって、サービス プロバイダーは加入者とのインタラクション数を制御できます。どの登録済みのデバイスについても、サービス プロバイダーはそのデバイスに対する変更があれば処理できるように準備する必要があります。背後に未登録のコンピュータがあるケーブル モデムを 100 台登録する作業と、それぞれの背後に登録済みコンピュータが大量に存在する可能性のあるケーブル モデムを 100 台登録する作業では、大きな違いがあります。この理由で、サービス プロバイダーは標準モード、無差別モード、ローミング モード、および混在モードから登録モードを慎重に選択する必要があります。

標準モード

標準モード（固定モードとも呼ばれる）での動作時、コンピュータは登録され、そのコンピュータが正しいケーブル モデムの背後にあれば、登録済みのアクセス権を受け取ります。このコンピュータが別のケーブル モデムの背後に移動すると、プロビジョニングされていないアクセス権が与えられます。

無差別モード

無差別モードでの動作時、DOCSIS モデムだけが登録され、別のデバイスの背後で動作しているデバイスに関するリース情報は DHCP サーバに保持されます。登録済みのデバイスの背後にある指定されたタイプのデバイスは、すべてネットワーク アクセス権を受け取ります。

ローミング モード

ローミング モードでの動作時、登録済みのデバイスは、他のどの登録済みデバイスの背後でも割り当てられたサービスを受け取ります。たとえば、このモードでは、ラップトップを使用しながら場所を移動し、複数のケーブル モデムからサービスを得ることができます。

混在モード

混在モードでの動作時、（複数のデバイスを持つ）1 つの配備内で、いつでも任意のモードが使用されます。

CPE プロビジョニング フロー

この項では、デバイスのプロビジョニング ワークフローについて説明します。

- 初期設定ワークフロー（P.4-10）
- 構成の更新ワークフロー（P.4-13）

初期設定ワークフロー

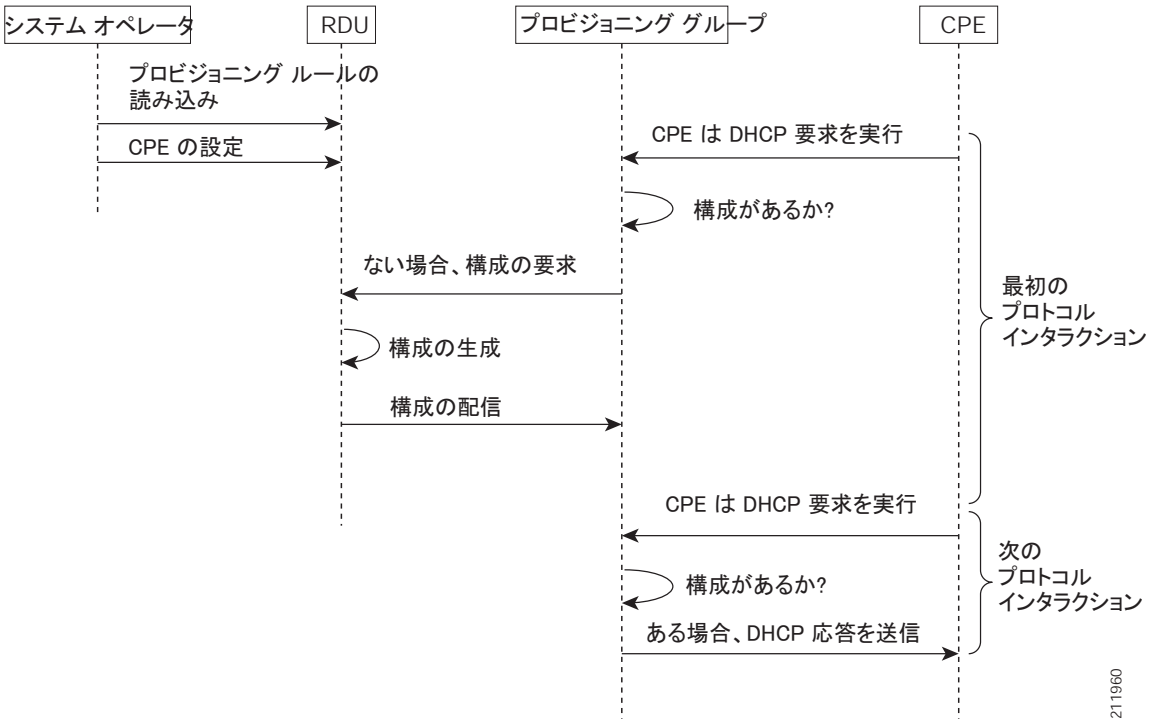
この項では、デバイスが最初にインストールされ、ブートされるときの設定フローについて説明します。このワークフローは、配備および登録モードによって異なります。次のワークフローがあります。

- 事前プロビジョニングされたデバイスのワークフロー（P.4-10）
- セルフプロビジョニングされたデバイスのワークフロー（P.4-12）

事前プロビジョニングされたデバイスのワークフロー

この項では、事前プロビジョニングされたデバイスのワークフローについて説明します。図 4-2 は、一般的な初期設定フローを示しています。

図 4-2　初期デバイス設定のワークフロー：事前プロビジョニング モード



1. BAC API から RDU に、さまざまなデバイス タイプ用に特別に定義された構成およびルールが読み込まれます。デバイスが事前設定され、サービス クラスに関連付けられ、RDU データベースに事前登録されます。

（注） CPE を事前設定すると、API を介して BAC にデバイス情報（MAC アドレス、サービスクラスなどの）が読み込まれます。
事前プロビジョニング モードでは、この処理が行われた後にネットワーク上でデバイスがブートします。セルフプロビジョニング モードでは、ネットワーク上でデバイスがブートしてからこの処理が行われます。

2. ブート時に、デバイスは自分のプロビジョニング グループを検出し、プロビジョニング グループの DPE に対して自動プロビジョニング フローを開始します。Cable Modem Termination System（CMTS; ケーブル モデム ターミネーション システム）がブロードキャスト トラフィックを DHCP サーバにリレーします。DHCP サーバまたは Network Registrar DHCP サーバ上の BAC 拡張が DPE に構成を要求します。

（注） デバイスがローミング モードを使用して新しいプロビジョニング グループに移動した場合は、前に属していたプロビジョニング グループから古い構成が削除される点を除き、同じフローがデバイスに対して実行されます。

3. DPE は、デバイス要求を受信すると、キャッシュにデバイスの構成があるかどうか検索します。デバイスはこれまでこのプロビジョニング グループと交信したことがないため、構成は見つかりません。次に、プロビジョニング グループ内の Network Registrar 拡張が、RDU にデバイスの構成を生成するように要求します。

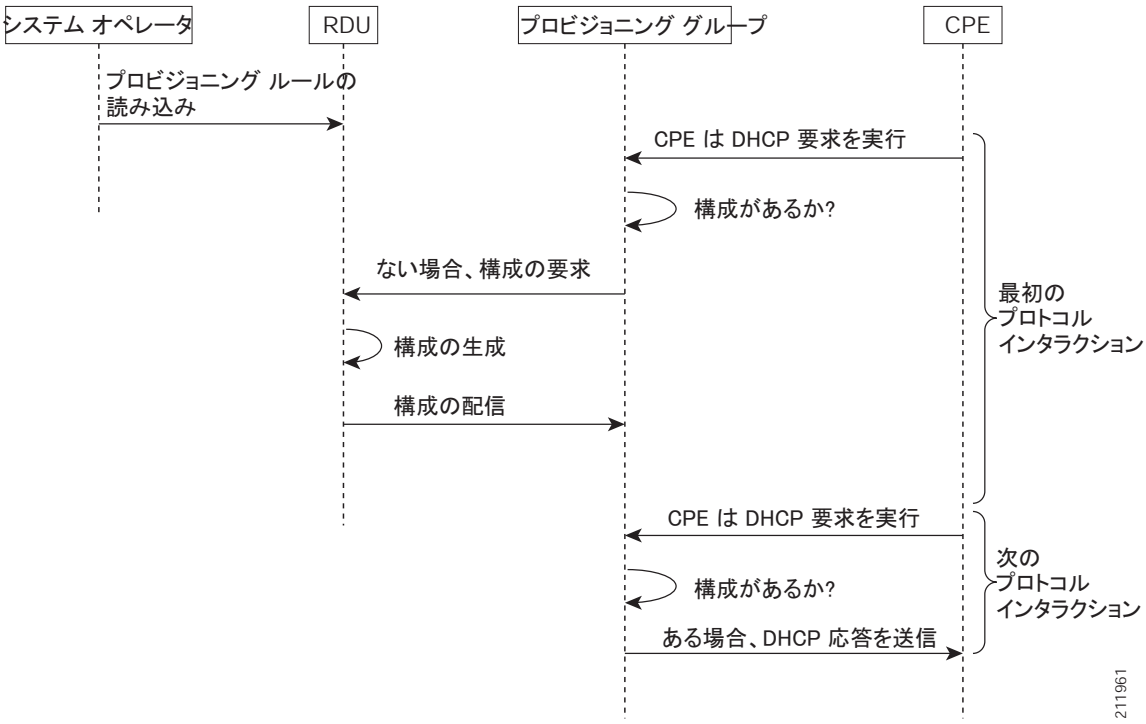
   RDU で要求処理にかかる時間によっては、プロビジョニング グループがデバイス要求に応答しないことを決定する場合もあります。

4. RDU が、デバイスに適した構成を生成します。生成されるデバイス構成では、DHCP 検出など、さまざまな CPE プロトコル イベントに対する DPE 応答が指定されます。
5. デバイス構成が DPE に転送され、そこでキャッシュされます。この段階で、DPE は、このデバイスに対する以後の CPE プロトコルのインタラクションを、RDU から独立して処理するようにプログラムされています。デバイスがネットワークに追加され、そのデバイスの構成が生成された場合、デバイスがブートすると、DPE は、事前登録されたデバイスとのインタラクションを開始できます。
6. デバイスとのインタラクションにおいて、追加情報を検出して RDU に転送することができます。この場合、RDU では、新しい構成を生成してすべての DPE に転送することがあります。

セルフプロビジョニングされたデバイスのワークフロー

この項では、セルフプロビジョニングされたデバイスのワークフローについて説明します。図 4-3 は、一般的な初期設定フローを示しています。

図 4-3　初期デバイス設定のワークフロー：セルフプロビジョニング モード



1. BAC API から RDU に、さまざまなデバイス タイプ用に特別に定義された構成およびルールが読み込まれます。

（注） CPE を事前設定すると、API を介して BAC にデバイス情報（MAC アドレス、サービス クラスなどの）が読み込まれます。セルフプロビジョニング モードでは、ネットワーク上でデバイスがブートしてからこの処理が行われます。

2. ブート時に、デバイスは自分のプロビジョニング グループを検出し、プロビジョニング グループの DPE に対して自動プロビジョニング フローを開始します。Cable Modem Termination System（CMTS; ケーブル モデム ターミネーション システム）がブロードキャスト トラフィックを DHCP サーバにリレーします。DHCP サーバまたは Network Registrar DHCP サーバ上の BAC 拡張が DPE に構成を要求します。

（注） デバイスがローミング モードを使用して新しいプロビジョニング グループに移動した場合は、前に属していたプロビジョニング グループから古い構成が削除される点を除き、同じフローがデバイスに対して実行されます。

3. DPE は、デバイス要求を受信すると、キャッシュにデバイスの構成があるかどうか検索します。デバイスはこれまでこのプロビジョニング グループと交信したことがないため、構成は見つかりません。次に、プロビジョニング グループ内の Network Registrar 拡張が、RDU にデバイスの構成を生成するように要求します。

   RDU で要求処理にかかる時間によっては、プロビジョニング グループがデバイス要求に応答しないことを決定する場合もあります。

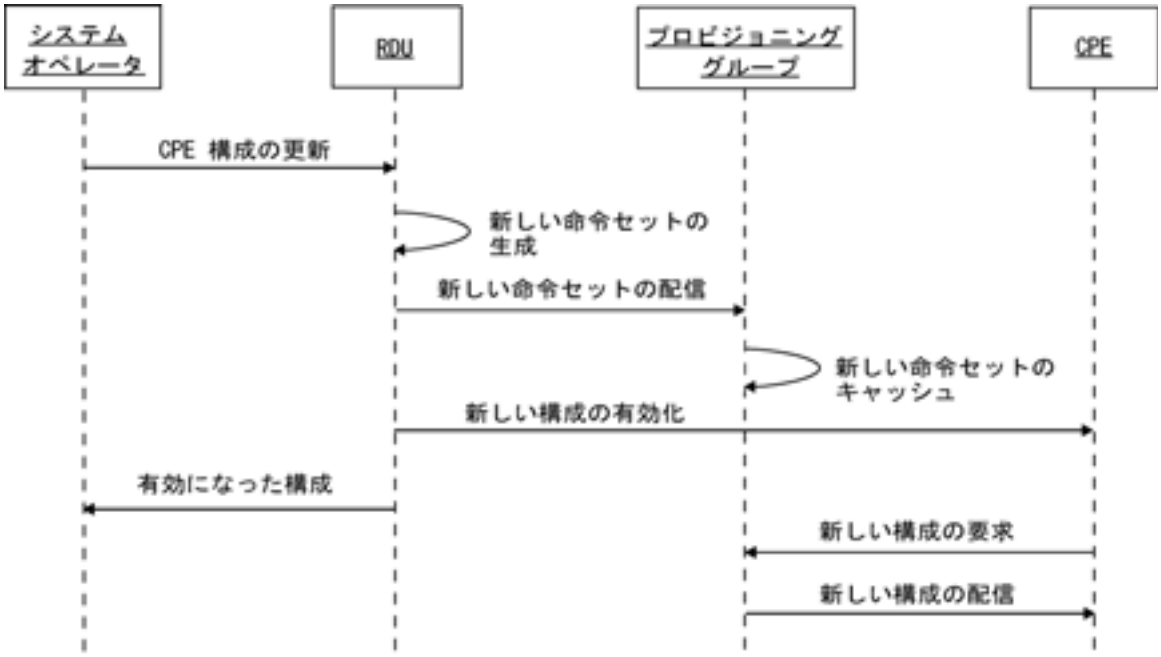
4. RDU が、デバイスに適した構成を生成します。生成される構成では、DHCP 検出など、さまざまな CPE プロトコル イベントに対する DPE 応答が指定されます。
5. デバイス構成が DPE に転送され、そこでキャッシュされます。この段階で、DPE は、このデバイスに対する以後の CPE プロトコルのインタラクションを、RDU から独立して処理するようにプログラムされています。デバイスがネットワークに追加され、そのデバイスの構成が生成された場合、デバイスがブートすると、DPE は、事前登録されたデバイスとのインタラクションを開始できます。
6. デバイスとのインタラクションにおいて、追加情報を検出して RDU に転送することができます。この場合、RDU では、新しい構成を生成してすべての DPE に転送することがあります。

構成の更新ワークフロー

この項では、デバイス構成が更新されたときのワークフローについて説明します。

図 4-4 に、以前に設定されたデバイスの構成を変更した場合の一般的な設定ワークフローを示します。

図 4-4　デバイス構成の更新ワークフロー



1. BAC API から、RDU のデバイス構成が更新されます。
2. RDU はデバイスの構成を生成して、デバイスが属するプロビジョニング グループの各 DPE に配信します。
3. DPE は新しい構成をキャッシュします。
4. RDU は、DPE に新しい構成をデバイスに転送するよう指示します。
5. ケーブルの場合、RDU はモデムまたは MTA で SNMP 設定を実行してデバイスをリブートします。

デバイスの無差別アクセス権

この項では、無差別アクセス権が与えられているデバイスの構成の制御に使用されるオブジェクトとプロパティについて説明します。

デバイスは、BAC に事前登録されていなくても、ブートと設定が許可されていれば、無差別アクセス権が与えられます。無差別アクセス権は通常、登録済みの DOCSIS モデムの背後にあるコンピュータなどのデバイスに使用されます。登録済みの DOCSIS モデムの背後にある不明なデバイスに対して無差別アクセス権がイネーブルになっていない場合、そのデバイスはデフォルトのサービス レベルを受け取ります。

デバイスに無差別アクセス権を付与するには、次の操作を実行する必要があります。

- 所定のタイプの不明なデバイスに対して、無差別ポリシーをイネーブルまたはディセーブルにします。無差別アクセス権がイネーブルになっているデバイスは、デフォルト構成を受け取るのではなく、ポリシーに従って設定されます。
- デバイスが無差別アクセス権の付与対象である場合、所定のタイプの不明なデバイスを対象としたサービス クラスを指定します。
- デバイスが無差別アクセス権の付与対象である場合、所定のタイプの不明なデバイスを対象とした DHCP 基準を指定します。

無差別アクセス権の設定

表 4-4 に、デバイスの無差別ポリシーを設定する方法を示します。

表 4-4　デバイスの無差別アクセス権の設定

設定のスコープ	使用する API コール
リレー エージェントのプロビジョニング グループ：たとえば、無差別アクセス権を、特定のプロビジョニング グループに含まれる登録済みのリレー エージェント デバイスの背後にあるコンピュータだけに許可するように設定できます。	*getProvGroupProperties* *changeProvGroupProperties*
リレー エージェントのサービス クラス オブジェクト：たとえば、無差別アクセス権を、特定のサービス クラスに関連付けられている DOCSIS モデムの背後にあるコンピュータだけに許可するように設定できます。	*addClassOfService* *changeClassOfServiceProperties* *getClassOfServiceProperties*
リレー エージェントの DHCP 基準：たとえば、無差別アクセス権を、特定の DHCP 基準に関連付けられている DOCSIS モデムの背後にあるコンピュータだけに許可するように設定できます。	*addDHCPCriteria* *changeDHCPCriteriaProperties* *getDHCPCriteriaDetails*
テクノロジー固有のデフォルト：たとえば、DOCSIS モデムのテクノロジー デフォルトを使用して、DOCSIS モデムの背後にあるコンピュータに対して無差別アクセス権を設定できます。	*changeDefaults* *getDefaults*
システム全体のデフォルト：グローバル システム デフォルト	*changeSystemDefaults* *getSystemDefaults*

（注） 特定のモデムなどのデバイスに、無差別ポリシーを直接設定することはできません。

無差別アクセス権とプロパティ階層

BAC では、複数のオブジェクトに対して無差別ポリシーを設定できます。したがって、優先される設定値を理解することが重要です。ポリシーがプロパティを使用して設定されるのに対し、プロパティの優先度は BAC プロパティ階層によって決まります。プロパティ階層内で特定のプロパティを持つ最初のオブジェクトによって、BAC が使用する値が決まります。

BAC は、デバイスのリレー エージェントのプロパティ階層内で無差別ポリシーのプロパティを検索します。たとえばコンピュータの場合、BAC は、コンピュータのリレーとして機能するケーブルモデムのプロパティ階層内で無差別ポリシーの設定値を検索します。プロパティ階層の詳細については、P.4-7 の「プロパティ階層」を参照してください。無差別ポリシーの詳細については、P.4-16 の「無差別ポリシー設定のプロパティ」を参照してください。

（注） テクノロジー デフォルトを使用して無差別アクセス権を設定すると、対象となるデバイス タイプではなく、リレー エージェントに関連付けられたオブジェクトにプロパティを設定する必要があります。たとえば、DOCSIS モデムの背後にあるコンピュータに対して無差別アクセス権をイネーブルにするには、コンピュータのテクノロジー デフォルトではなく、その DOCSIS モデムのテクノロジー デフォルトへの無差別アクセス権をイネーブルにします。

無差別ポリシーのプロパティではデバイス タイプごとに、サービス クラスと DHCP 基準を指定し、また無差別アクセス権をイネーブルまたはディセーブルに指定します。デバイスで無差別モードがイネーブルになっているにもかかわらず、デバイスのリレー エージェント階層の検索で、サービス クラスまたは DHCP 基準のプロパティに一致するものが検出されない場合、非無差別アクセス権のデフォルトのサービス クラスまたは DHCP 基準が使用されます。たとえば、BAC がコンピュータに無差別アクセス権を付与するように設定されているが、無差別のサービス クラスまたは DHCP 基準（あるいはその両方）を見つけられない場合、コンピュータに対してデフォルトのサービス クラス、DHCP 基準、またはその両方が使用されます。

サービス クラスと DHCP 基準のデフォルトは、次のプロパティを使用して、対象デバイス（リレー エージェントではなく）のテクノロジー デフォルトに設定されます。

- サービス クラス：*/default/classOfService*

  API 定数は `TechnologyDefaultsKeys.DEFAULT_CLASS_OF_SERVICE` です。
- DHCP 基準：*/default/dhcpCriteria*

  API 定数は `TechnologyDefaultsKeys.DEFAULT_DHCP_CRITERIA` です。

無差別モードのデバイス用構成の生成

無差別モードのデバイスの構成は、次の条件で生成されます。

- デバイスが最初にオンラインになり、無差別アクセス権が与えられた。
- 最新ではなくなった DPE がキャッシュを読み込み中で、特定のプロビジョニング グループの構成を要求した。
- API コール *regenConfigs* でデバイス構成の再生成が明示的に要求された。
- 無差別アクセス権を持つデバイスのリレー エージェント デバイスの構成が再生成中である。
- 無差別ポリシー（またはその他の設定）が変更されたために、BAC Configuration Regeneration Service（CRS; 構成再生成サービス）が影響を受けるデバイスの構成を再生成する必要が生じた。

無差別モードのデバイスの構成が再生成されるたびに、新しく設定された無差別ポリシーが使用されます（たとえば、無差別モードのコンピュータに現在指定されているサービス クラス）。ただし、デバイスが無差別モードのデバイスとしてオンラインになった後に API を介してそのデバイスのサービス クラスまたは DHCP 基準が変更された場合、それ以降、そのデバイスは無差別とは見なされず、無差別ポリシーに変更を加えても影響を受けません。デバイスはそれ以降、登録済みと見なされます。

無差別ポリシー設定のプロパティ

デバイスに無差別アクセス権を設定するには、BAC がサポートする特定のデバイス タイプに関連付けられたプロパティを設定する必要があります。デバイス タイプに対して無差別アクセス権をイネーブルまたはディセーブルにできます。

- Enabled：表 4-4 に示す API コールに関連付けられたスコープ内で、デバイスの無差別アクセス権をイネーブルにします。
- Disabled：無差別アクセス権をディセーブルにします。プロパティが存在しない場合、Disabled 設定がデフォルトになります。

無差別アクセス権を設定するプロパティのリストについては、表 4-5 を参照してください。

無差別ポリシーのプロパティは、読み取り / 書き込みプロパティと読み取り専用のプロパティに分かれます。この項では、読み取り / 書き込みプロパティと読み取り専用プロパティについて説明します。これらのプロパティには、デバイスの無差別アクセス権をイネーブルにするために設定が必要なものと、デバイスのサービス クラスと DHCP 基準を選択するために設定するものがあります。

読み取り / 書き込みプロパティ

（注） この項で説明するすべてのプロパティに適用可能な API コールについては、表 4-4 を参照してください。

表 4-5 に、無差別アクセス権をイネーブルにするために使用できるプロパティを示します。

表 4-5　無差別アクセス権をイネーブルにするためのプロパティ

プロパティ名	説明
/promiscuousMode/enable/Computer	リレー エージェントのプロパティ階層に、「true」または「false」のブール値を次のように設定します。 • true：該当するリレーの背後にあるコンピュータの無差別アクセス権をイネーブルにします。 • false：該当するリレーの背後にあるコンピュータの無差別アクセス権をディセーブルにします。 プロパティがリレー エージェントのプロパティ階層にない場合、該当するリレーの背後にあるコンピュータの無差別アクセス権は許可されず、デバイスはデフォルト アクセス権を受け取ります。
	API 定数 `PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED`

表 4-5　無差別アクセス権をイネーブルにするためのプロパティ（続き）

プロパティ名	説明
/promiscuousMode/enable/PacketCableMTA	リレー エージェントのプロパティ階層に、「true」または「false」のブール値を次のように設定します。 • true：該当するリレーの背後にある PacketCable MTA の無差別アクセス権をイネーブルにします。 • false：該当するリレーの背後にある PacketCable MTA の無差別アクセス権をディセーブルにします。 プロパティがリレー エージェントのプロパティ階層にない場合、該当するリレーの背後にある PacketCable MTA の無差別アクセス権は許可されず、デバイスはデフォルト アクセス権を受け取ります。
	API 定数 `PolicyKeys.PACKET_CABLE_MTA_PROMISCUOUS_MODE_ENABLED`
/promiscuousMode/enable/STB	リレー エージェントのプロパティ階層に、「true」または「false」のブール値を次のように設定します。 • true：該当するリレーの背後にある STB の無差別アクセス権をイネーブルにします。 • false：該当するリレーの背後にある STB の無差別アクセス権をディセーブルにします。 プロパティがリレー エージェントのプロパティ階層にない場合、該当するリレーの背後にある STB の無差別アクセス権は許可されず、デバイスはデフォルト アクセス権を受け取ります。
	API 定数 `PolicyKeys.STB_PROMISCUOUS_MODE_ENABLED`
/promiscuousMode/enable/CableHomeWanData	リレー エージェントのプロパティ階層に、「true」または「false」のブール値を次のように設定します。 • true：該当するリレーの背後にある CableHome WAN-Data デバイスの無差別アクセス権をイネーブルにします。 • false：該当するリレーの背後にある CableHome WAN-Data デバイスの無差別アクセス権をディセーブルにします。 プロパティがリレー エージェントのプロパティ階層にない場合、該当するリレーの背後にある WAN-Data デバイスの無差別アクセス権は許可されず、デバイスはデフォルト アクセス権を受け取ります。
	API 定数 `PolicyKeys.CABLE_HOME_WAN_DATA_PROMISCUOUS_MODE_ENABLED`

表 4-5　無差別アクセス権をイネーブルにするためのプロパティ（続き）

プロパティ名	説明
/promiscuousMode/enable/CableHome WanMan	リレー エージェントのプロパティ階層に、「true」または「false」のブール値を次のように設定します。 • true：該当するリレーの背後にある CableHome WAN-MAN デバイスの無差別アクセス権をイネーブルにします。 • false：該当するリレーの背後にある CableHome WAN-MAN デバイスの無差別アクセス権をディセーブルにします。 プロパティがリレー エージェントのプロパティ階層にない場合、該当するリレーの背後にある WAN-MAN デバイスの無差別アクセス権は許可されず、デバイスはデフォルト アクセス権を受け取ります。
	API 定数 `PolicyKeys.CABLE_HOME_WAN_MAN_PROMISCUOUS_MODE_ENABLED`
/promiscuousMode/enable/	このプロパティを使用して新しいタイプのデバイスに対して無差別アクセス権をイネーブルまたはディセーブルにするには、プロパティ名に有効なデバイス タイプの名前を付加します。 「true」または「false」のブール値を設定します。
	API 定数 `PolicyKeys.PROMISCUOUS_MODE_PREFIX`

表 4-6 に、無差別アクセス権を付与されたデバイスのサービス クラスを選択するために設定が必要な読み取り / 書き込みプロパティを示します。

表 4-6　無差別アクセス権：サービス クラスの読み取り / 書き込みプロパティ

サービス クラスのプロパティ名	説明
/provisioning/cpeClassOfService/Computer	無差別モードのコンピュータに対して選択される既存のサービス クラスの名前を指定します。
	API 定数 `PolicyKeys.COMPUTER_CLASS_OF_SERVICE`
/provisioning/cpeClassOfService/Packet CableMTA	無差別モードの PacketCable MTA に対して選択される既存のサービス クラスの名前を指定します。
	API 定数 `PolicyKeys.PACKET_CABLE_MTA_CLASS_OF_SERVICE`
/provisioning/cpeClassOfService/STB	無差別モードのセットトップ ボックスに対して選択される既存のサービス クラスの名前を指定します。
	API 定数 `PolicyKeys.STB_CLASS_OF_SERVICE`

表 4-6　無差別アクセス権：サービス クラスの読み取り / 書き込みプロパティ（続き）

サービス クラスのプロパティ名	説明
/provisioning/cpeClassOfService/CableHome WanMan	無差別モードの CableHome WAN-Data デバイスに対して選択される既存のサービス クラスの名前を指定します。
	API 定数 `PolicyKeys.CABLEHOME_WAN_DATA_CLASS_OF_SERVICE`
/provisioning/cpeClassOfService/CableHome WanData	無差別モードの CableHome WAN-MAN デバイスに対して選択される既存のサービス クラスの名前を指定します。
	API 定数 `PolicyKeys.CABLEHOME_WAN_MAN_CLASS_OF_SERVICE`
/provisioning/cpeClassOfService/	指定したデバイス タイプのデバイスに対して選択される既存のサービス クラスを指定します。このプロパティ名は、有効なデバイス タイプ名と一緒に使用します。このプロパティは、カスタム デバイス タイプに使用できます。
	API 定数 `PolicyKeys.PROMISCUOUS_COS_PREFIX`

表 4-7 に、無差別アクセス権を付与されたデバイスの DHCP 基準を選択するために設定が必要な読み取り / 書き込みプロパティを示します。

表 4-7　無差別アクセス権：DHCP 基準の読み取り / 書き込みプロパティ

DHCP 基準のプロパティ名	説明
/provisioning/cpeDhcpCriteria/Computer	無差別モードのコンピュータに対して選択される既存の DHCP 基準オブジェクトの名前を指定します。
	API 定数 `PolicyKeys.COMPUTER_DHCP_CRITERIA`
/provisioning/cpeDhcpCriteria/Packet CableMTA	無差別モードの PacketCable MTA に対して選択される既存の DHCP 基準オブジェクトの名前を指定します。
	API 定数 `PolicyKeys.PACKET_CABLE_MTA_DHCP_CRITERIA`
/provisioning/cpeDhcpCriteria/STB	無差別モードのセットトップ ボックスに対して選択される既存の DHCP 基準オブジェクトの名前を指定します。
	API 定数 `PolicyKeys.STB_ DHCP_CRITERIA`
/provisioning/cpeDhcpCriteria/CableHome WanData	無差別モードの CableHome WAN-Data デバイスに対して選択される既存の DHCP 基準オブジェクトの名前を指定します。
	API 定数 `PolicyKeys.CABLEHOME_WAN_DATA_ DHCP_CRITERIA`

表 4-7　無差別アクセス権：DHCP 基準の読み取り / 書き込みプロパティ（続き）

DHCP 基準のプロパティ名	説明
/provisioning/cpeDhcpCriteria/CableHomeWanMan	無差別モードの CableHome WAN-MAN デバイスに対して選択される既存の DHCP 基準オブジェクトの名前を指定します。
	API 定数 `PolicyKeys.CABLEHOME_WAN_MAN_ DHCP_CRITERIA`
/provisioning/cpeDhcpCriteria/	指定したデバイス タイプのデバイスに対して選択される既存の DHCP 基準オブジェクトを指定します。このプロパティ名は、有効なデバイス タイプ名と一緒に使用します。このプロパティは、カスタム デバイス タイプに使用できます。
	API 定数 `PolicyKeys.PROMISCUOUS_DC_PREFIX`

読み取り専用プロパティ

表 4-8 に、デバイスのサービス クラスと DHCP 基準を選択するために設定が必要な読み取り専用の無差別プロパティを示します。これらの読み取り専用プロパティは、前の項で指定した読み取り / 書き込みプロパティと合せて、現在のシステム構成を判別するのに役立ちます。

表 4-8　無差別アクセス権：読み取り専用プロパティ

プロパティ名	説明	
/isSystemWide/default/promiscuous	所定のサービス クラス オブジェクトまたは DHCP 基準オブジェクトが、無差別モードのデバイスのシステム全体のデフォルトとして参照される場合、値「true」を返します。	
	適用可能な API コール *getClassOfServiceProperties* *getDHCPCriteriaDetails*	**API 定数** `PolicyKeys.IS_SYSTEM_WIDE_ DEFAULT_PROMISCUOUS`
/referencedBy/deviceTypes/forPromiscuousDevices	無差別ポリシーのプロパティ内にある、所定のサービス クラス オブジェクトまたは DHCP 基準オブジェクトを参照するデバイス タイプ オブジェクト（テクノロジー）名のリストを返します。	
	適用可能な API コール *getClassOfServiceProperties* *getDHCPCriteriaDetails*	**API 定数** `PolicyKeys.REFERENCED_BY_ DEVICE_TYPE_FOR_PROMISCUOUS_ DEVICES`

表 4-8　無差別アクセス権：読み取り専用プロパティ（続き）

プロパティ名	説明	
/related/classesOfService	無差別ポリシーのプロパティ内にある、所定のサービス クラス オブジェクトまたは DHCP 基準オブジェクトが使用するサービス クラス オブジェクト名のリストを返します。 （注）このプロパティは、サービス クラス リストを取得するためのショートカットとして使用できます。リストは、このオプションに設定されている個々の無差別ポリシー プロパティを読み込んで取得することもできます。	
	適用可能な API コール *getClassOfServiceProperties* *getDHCPCriteriaDetails*	**API 定数** `PolicyKeys.RELATED_CLASS_OF_SERVICE`
/related/dhcpCriteria	無差別ポリシーのプロパティ内にある、所定のサービス クラス オブジェクトまたは DHCP 基準オブジェクトが使用する DHCP 基準オブジェクト名のリストを返します。 （注）このプロパティは、DHCP 基準リストを取得するためのショートカットとして使用できます。リストは、このオプションに設定されている個々の無差別ポリシー プロパティを読み込んで取得することもできます。	
	適用可能な API コール *getClassOfServiceProperties* *getDHCPCriteriaDetails*	**API 定数** `PolicyKeys.RELATED_DHCP_CRITERIA`

無差別モードのデバイスのカスタム ポリシー

前の項で指定したプロパティを使用して、デバイスの無差別ポリシーを設定できます。ただし、追加のロジックが必要な場合は、拡張とカスタム プロパティを使用してカスタム ロジックを実装できます。カスタム プロパティを使用すると、新しいプロパティを定義できます。定義されたプロパティは、API を介して任意のオブジェクト上に格納することができます。

無差別モードのデバイスのポリシーを強化するために、次の拡張を使用できます。

- デバイス検出：デバイスのテクノロジー タイプを判別します（通常は DHCP 要求データに基づく）。この拡張が検出した情報は、Device Detection Context に追加され、他の拡張から使用できるようになります。
- サービス レベル選択：デバイスに対して適切なサービス クラス オブジェクトおよび DHCP 基準オブジェクトを選択します。無差別ポリシーのプロパティによって、無差別アクセス権を持つデバイスのサービス クラスおよび DHCP 基準が決まります。
- 構成生成：デバイスと、必要に応じてその背後にあるデバイスの構成を生成します。サービス レベル選択拡張が選択したポリシーに基づいて、リレー エージェントの背後にある無差別モードのデバイスの構成が再生成されます。リレー エージェントの背後にあるデバイスの構成を再生成するデフォルトの動作を強化する場合だけ、拡張の変更が必要になることがあります。

CHAPTER 5

設定テンプレートの管理

この章では、Broadband Access Center（BAC）でサポートされる、デバイス構成およびデバイス管理用のテンプレートについて説明します。この章は、次の項で構成されています。

テンプレート ファイル：概要

BAC が使用するテンプレートは、動的 PacketCable、DOCSIS、および CableHome ファイルを配備するときに役立ちます。テンプレートを使用して、読みやすい形式のテンプレート ファイルを作成し、迅速かつ簡単に編集することができます。テンプレートは、有効な PacketCable、DOCSIS、または CableHome ファイルを生成するときに使用する PacketCable、DOCSIS、または CableHome のオプションおよび値を表す ASCII テキスト ファイルです。BAC は *.tmpl* 拡張子を使用してテンプレート ファイルを識別します。サービス クラスでテンプレート ファイルを参照する前に、管理者のユーザ インターフェイスまたは Application Programming Interface（API; アプリケーション プログラミング インターフェイス）を使用して、ファイルとしてそのテンプレート ファイルを RDU に追加する必要があります。

BAC RDU コンポーネントをインストールするときに、いくつかのサンプル テンプレート ファイルが *BPR_HOME/rdu/templates* ディレクトリにコピーされます。

テンプレートを作成または編集するために必要なツールは単純なテキスト エディタだけですが、独自のテンプレート ファイルを作成する前に、次の情報を十分に理解しておく必要があります。

- BAC プロビジョニングのフロー
- DOCSIS 1.0、1.1、2.0、および 3.0 RFI の仕様
- DOCSIS レイヤ 2 バーチャル プライベート ネットワークの仕様
- PacketCable 1.0、1.1、および 1.5 の仕様
- Multimedia Terminal Adapter（MTA; マルチメディア ターミナル アダプタ）デバイスのプロビジョニングの仕様
- CableHome 1.0 の仕様
- ケーブル デバイス用の SNMP MIB（DOCS-CABLE-DEVICE-MIB など）

テンプレート文法

テンプレートは、次の種類の文で構成されます。

- コメント（P.5-3）
- インクルード（P.5-3）
- オプション（P.5-4）
- インスタンス修飾子（P.5-5）
- OUI 修飾子（P.5-5）

comment 文で、テンプレートに説明を入れることができます。include 文で、他のテンプレートで使用するビルディング ブロック テンプレートを作成できます。オプションを使用して、説明のために PacketCable、DOCSIS、または CableHome の Type Length Value（TLV）を指定します。インスタンス修飾子を使用して、複合オプションを特定の個別 TLV にグループ化できます。OUI 修飾子を使用すると、ベンダー固有の情報を含めることができます。表 5-1 に、利用可能なテンプレート文法のオプションを示します。

表 5-1　テンプレートの文法

オプション	説明
<comment>	::= #[ascii-string]
<include>	::= include "<filename.tmpl>"
<option-description>	::= option <option-num> [instance <instance-num>] [oui <oui>] <option-value>
<option-num>	::= <unsigned-byte>[.<unsigned-byte>]*

表 5-1　テンプレートの文法（続き）

オプション	説明
<option-value>	::= <well-defined-value> | <custom-value>
<well-defined-value>	::= <option-value-string>[,<option-value-string>]*
<custom-value>	::= <ascii-value> | <hex-value> | <ip-value> | <snmp-value>
<ascii-value>	::= ascii <ascii-string>
<hex-value>	::= hex <hex-string>
<ip-value>	::= ip <ip-string>
<instance-num>	::= <unsigned integer>
<template>	::= <template-statement>*
<template-statement>	::= <comment> | <include> | <option-description>
<snmp-value>	::= <snmpvar-oid>,<snmpvar-type>,<snmpvar-value>

コメント

コメントは情報のみを提供し、常にシャープ記号（#）から行末までの間に配置されます。例 5-1 に、コメントの使用方法の例を示します。

例 5-1　コメントの使用方法の例

```
#
# Template for gold service
#

option 3 1 # enabling network access
```

インクルード

インクルード ファイルを使用すると、似ているが少し異なるテンプレートの階層を構築できます。これは、多くのサービス クラスで共通のオプションを定義する場合に、複数のテンプレートでオプションを重複させる必要がなくなり、とても便利です。

単一のテンプレートで複数の include 文を使用できますが、テンプレートの中での include 文の位置は重要です。インクルード ファイルの内容は、テンプレートの中に include 文が見つかった場所で読み込まれます。インクルードするテンプレートは、使用する前にファイルとして RDU に追加する必要があります。テンプレートは RDU データベースにパス情報がない状態で格納されるため、インクルードするファイルには ../.. などの位置修飾子を含めることができません。例 5-2 と例 5-3 は、それぞれ、インクルード オプションの正しい使用方法と誤った使用方法を示しています。

例 5-2　include 文の正しい使用方法

```
# Valid, including common options
include "common_options.tmpl"
```

例 5-3　include 文の誤った使用方法

```
# Invalid, using location modifier
include "../common_options.tmpl"

# Invalid, using incorrect file suffix
include "common_options.common"

# Invalid, not using double quotes
include common_options.tmpl
```

オプション

PacketCable、DOCSIS、および CableHome の設定ファイルは、適切に符号化されたオプション ID と値のペアで構成されます。サポートされるオプションの形式には、定義済みとカスタムの 2 種類があります。

- オプションを正しく定義するには、オプション番号と値が必要です。値は、オプション番号の符号化タイプに基づいて符号化されます。
- カスタム オプションには、オプション番号、明示的な値の符号化タイプ、および値が必要です。

Option 43 などの複合オプションを使用するときは、インスタンス修飾子を使用して、TLV グループを指定できます。P.5-5 の「インスタンス修飾子」を参照してください。

テンプレートで定義済みオプションのいずれかを指定するときは、値の値符号化を指定する必要はありません。定義済み符号化タイプの詳細については、P.5-15 の「定義済みオプションの符号化タイプ」および P.B-2 の「DOCSIS オプションのサポート」を参照してください。

カスタム オプション（Option 43 など）を指定するときは、オプションの符号化タイプを指定する必要があります。利用可能な符号化タイプは、次のとおりです。

- ASCII：ASCII タイプは、指定された任意の値を NULL ターミネータなしの ASCII 文字列として符号化します。値にスペースを含める場合は、二重引用符で囲む必要があります。
- hex：値は有効な 16 進数で、各オクテットに正確に 2 文字入るようにする必要があります。値として 01 を指定した場合は、符号化で正確に 1 オクテットが使用されます。値として 0001 を指定した場合は、符号化で正確に 2 オクテットが使用されます。
- IP アドレス：IP アドレス タイプでは、指定された任意の値が 4 オクテットに符号化されます。たとえば、IP アドレス 10.10.10.1 は 0A0A0A01 に符号化されます。
- SNMPVarBind：SNMP OID 文字列、タイプ、および値。これらはそれぞれカンマで区切られます。

1 行に複数の値があるオプションでは、カンマを使用して値を区切ります。それぞれの値は別個に扱われるので、場合によっては値の 1 つを二重引用符で囲む必要がありますが、他の値を囲む必要はありません。複数の値を持つオプションの例として、Option 11（SNMP VarBind）があります。詳細については、P.5-7 の「SNMP VarBind」を参照してください。

複合オプションを指定するときは、トップ レベル オプション（Option 4.1 を指定するときの Option 4 など）を指定する必要がありません。例 5-4 と例 5-5 は、それぞれ option 文の正しい使用方法と誤った使用方法を示しています。

例 5-4　option 文の正しい使用方法

```
# Valid, specifying the number for well known option 3
option 3 1

# Valid, specifying the number for option 4 sub-option 1
option 4.1 1

# Valid, specifying a vendor option as hex
option 43.200 hex 00000C

# Valid, specifying a vendor option as ascii
option 43.201 ascii "enable log"

# Valid, specifying a vendor option as IP
option 43.202 ip 10.4.2.1
```

例 5-5　option 文の誤った使用方法

```
# Invalid, using hex with incorrect hex separator
option 43.200 hex 00.00.0C

# Invalid, not using double quotes when needed
option 43.201 ascii enable log

# Invalid, not specifying IP address correctly
option 43.202 ip 10-10-10-1

# Invalid, specifying the description for option "Network Access Control"
option "Network Access Control" 1

# Invalid, specifying top level option
option 4
```

インスタンス修飾子

インスタンス修飾子は、複合オプションを特定の個別 TLV にグループ化するために使用します。例 5-6 と例 5-7 は、それぞれ、個別の TLV を作成する正しい方式と誤った方式を示しています。IOS コマンドを 2 つの個別のコマンドとして解釈するために、IOS DOCSIS モデムをイネーブルにする必要があります。

例 5-6　正しい IOS コマンドライン入力

```
# Valid, each IOS command gets its own TLV
option 43.8 instance 1 00-00-0C
option 43.131 instance 1 ascii "login"
option 43.8 instance 2 00-00-0C
option 43.131 instance 2 ascii "password cable"
```

例 5-7　誤った IOS コマンドライン入力

```
# Invalid, IOS commands are grouped into one TLV
option 43.8 00-00-0C
option 43.131 ascii "login"
option 43.131 ascii "password cable"

# Invalid, using instance on non-compound options
option 3 instance 1 1
```

（注） Option 43.8 の符号化タイプは、Organizationally Unique Identifier（OUI）です。例 5-4 に示されている例とは異なり、このタイプは 00-00-0C 形式のみ受け入れます。

OUI 修飾子

OUI 修飾子は、Option 43 とそのサブオプションを使用して、マルチベンダーのサポートを強化します。

BAC 4.0 では、単一のテンプレートを使用して、多くのベンダーのさまざまな TLV 43 を指定できます。例 5-8 は、OUI 形式を XX-XX-XX として指定します。

- FF-FF-FF：DOCSIS 一般拡張の符号化を指定するベンダー ID を示します。
- 00-00-0C：シスコ固有のケーブル モデム Option 43 とそのサブオプションを指定するシスコ ベンダー ID を示します。

例 5-8 は、L2VPN のアップストリーム トラフィックを分類するためにケーブル モデム設定ファイルを使用する、L2VPN の BAC サポートを示しています。このテンプレートの内容を使用して、次のサブ TLV を生成できます。

- OUI=FF-FF-FF を使用する、DOCSIS 一般拡張の符号化の 43.5.1 と 43.5.2.2。
- OUI=00-00-0C を使用する、シスコ固有の Option 43 の 43.1。

ただし、DOCSIS の仕様に準拠するために、TLV 43 の最初の サブ TLV として次のいずれかを挿入する必要があります。

- 一般拡張情報を符号化するために DOSCIS 拡張フィールドを使用する場合は、0xFFFFFF。
- シスコ固有の subTLV を生成する場合は、0x00000C。

例 5-8　OUI 修飾子の正しい使用方法

```
# Upstream L2VPN Classifier Example

# This example shows how to classify upstream traffic from a specific CPE
# onto an upstream L2VPN service flow, in which other CPE attached to
# the cable modem forward to the non-L2VPN forwarder, as depicted below.

# This example also demonstrates that when using the DOCSIS extension
# field (TLV 43) to encode general extension information (GEI), you do
# not need to specify oui=FF-FF-FF. You only need to specify the OUI tag when
# general extension encoding is not used and vendor-specific encoding is used.

# Upstream L2VPN Classifier Cable Modem Config File

# (43) Per-CM L2VPN Encoding
# GEI (43.8) Vendor ID : 0xFFFFFF for GEI
option 43.8 instance 1 ff-ff-ff

# GEI (43.5) for L2VPN Encoding
# GEI (43.5.1) VPNID Subtype
option 43.5.1  instance 1 0234560003

# GEI (43.5) for L2VPN Encoding
# GEI (43.5.2) IEEE 802.1Q Format Subtype
# VLAN ID 25
option 43.5.2.2 instance 1 25

# Cisco Specific Vendor Option Encodings
# (43.8) Vendor ID : 00-00-0C (Cisco Vendor ID)
option 43.8 instance 2 00-00-0C

# Cisco Vendor Specific option (43.1)
# Static Downstream Frequency
# Frequency 402750000
option 43.1 instance 2 oui 00-00-0C 402750000

# Cisco Specific Vendor Option Encodings
# (43.8) Vendor ID : 00-00-0C (Cisco Vendor ID)
option 43.8 instance 3 00-00-0C

# Cisco Vendor Specific option (43.3)
# Update Boot Monitor Image
# image name (boot_monitor_image.bin)
option 43.3 instance 3 oui 00-00-0C boot_monitor_image.bin
```

例 5-9 と例 5-10 は、OUI 修飾子の誤った使用方法を示しています。

例 5-9　OUI 修飾子の誤った使用方法

```
# Invalid, OUI tag needs to be present for each 43 suboption if/when general extension
# encoding is not used and vendor-specific encoding is used.

option 43.8 00-00-0C

option 43.3 boot_monitor_image.bin
```

例 5-10　OUI 修飾子の誤った使用方法

```
# Invalid, when both OUI and instance modifier are used in authoring a template,
# "instance" modifier needs to occur before "oui" modifier.

option 43.8 instance 1 00-00-0C

option 43.3 oui 00-00-0C instance 1 boot_monitor_image.bin
```

SNMP VarBind

DOCSIS Option 11、PacketCable Option 64、または CableHome Option 28 を指定するときは、Object Identifier（OID）を使用する必要があります。OID を含む MIB は、RDU がロードする次の MIB のいずれかに存在する必要があります。オブジェクトを一意に識別するために必要な数の OID を指定する必要があります。OID の名前または番号を使用できます。RDU は、次の MIB を自動的にロードします。

- SNMPv2-SMI
- SNMPv2-TC
- CISCO-SMI
- CISCO-TC
- SNMPv2-MIB
- RFC1213-MIB
- IANAifType-MIB
- IF-MIB

DOCSIS MIB

これらの DOCSIS MIB は、次の RDU にロードされます。

- DOCS-IF-MIB
- DOCS-BPI-MIB
- CISCO-CABLE-SPECTRUM-MIB
- CISCO-DOCS-EXT-MIB
- SNMP-FRAMEWORK-MIB
- DOCS-CABLE-DEVICE-MIB
- DOCS-CABLE-DEVICE-MIB-OBSOLETE
- CISCO-CABLE-MODEM-MIB

RDU にロードされる DOCS-CABLE-DEVICE MIB には、2 つのバージョンがあります。

- DOCS-CABLE-DEVICE-MIB-OBSOLETE（実験的ブランチ）
- DOCS-CABLE-DEVICE-MIB（mib2 ブランチ）

完全修飾された MIB OID（.experimental...）は常に、MIB OID を一意に識別します。

DOCS-CABLE-DEVICE-MIB のうち、完全修飾されていない MIB OID を使用する場合、MIB OID は常に、デフォルトで DOCS-CABLE-DEVICE-MIB に設定されます（DOCS-CABLE-DEVICE-MIB-OBSOLETE には設定されません）。

例 5-11 と例 5-12 は、それぞれ、完全修飾された MIB OID と完全修飾されていない MIB OID の使用方法を示しています。

例 5-11　完全修飾された MIB OID

```
# Valid, uniquely identifying an OID
option 11 .experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccess
Entry.docsDevNmAccessStatus.1, Integer, 4
```

例 5-12　完全修飾されていない MIB OID（デフォルトで DOCS-CABLE-DEVICE-MIB に設定される）

```
# Valid, NonFully Qualified MIB OID.
option 11 .docsDevNmAccessStatus.1, Integer, 4
```

配備中の DOCSIS CM が DOCS-CABLE-DEVICE-MIB-OBSOLETE を要求しない場合は、常に短縮形の MIB OID を使用できます。

PacketCable MIB

次の PacketCable（北米版）MIB が RDU にロードされます。

- CLAB-DEF-MIB
- PKTC-MTA-MIB
- PKTC-SIG-MIB
- PKTC-EVENT-MIB

CableHome MIB

次の CableHome MIB が RDU にロードされます。

- CABH-CAP-MIB
- CABH-CDP-MIB
- CABH-CTP-MIB
- CABH-PS-DEV-MIB
- CABH-QOS-MIB
- CABH-SEC-MIB

次の追加 MIB が必要ですが、BAC 製品の一部ではありません。

- CABH-CTP-MIB には RMON2-MIB、TOKEN-RING-RMON-MIB が必要です。
- CABH-SEC-MIB には DOCS-BPI2-MIB が必要です。

マクロ変数

マクロ変数はテンプレートで値として指定され、これを使用してデバイス固有のオプション値を指定できます。マクロ変数がテンプレートにあると、プロパティ階層でマクロ変数名が検索され、変数の値が代入されます。変数名はカスタム プロパティで、事前に RDU に定義します。スペースは使用できません。

（注） マクロ変数のカスタム プロパティを使用する場合は、DataType.STRING を使用する必要があります。

カスタム プロパティを定義すると、次のプロパティ階層で使用できるようになります。

- デバイス プロパティ
- プロビジョニング グループ プロパティ
- サービス クラス プロパティ
- DHCP 基準プロパティ
- テクノロジー デフォルト（PacketCable、DOCSIS、CableHome など）
- システム デフォルト

テンプレート パーサーは、階層の下から上にプロパティを検索し（最初はデバイス、次にサービス クラス）、テンプレート オプション構文に変換します。マクロ変数をサポートする構文は次のとおりです。

- ${var-name}：この構文は、単純な代入です。変数が見つからない場合、パーサーはエラーを生成します。
- ${var-name, ignore}：この構文では、変数値がプロパティ階層で見つからなかった場合、テンプレート パーサーはこのオプションを無視します。
- ${var-name, default-value}：この構文では、変数がプロパティ階層で見つからなかった場合に、デフォルト値が使用されます。

例 5-13 と例 5-14 は、それぞれ Option 11 の正しい使用方法と誤った使用方法を示しています。

例 5-13　マクロ変数の正しい使用方法

```
# Valid, using macro variable for max CPE's, straight substitution
option 18 ${MAX_CPES}

# Valid, using macro variable for max CPE's, ignore option if variable not found
# option 18 will not be defined in the DOCSIS configuration file if MAX_CPES
# is not found in the properties hierarchy
option 18 ${MAX_CPES, ignore}

# Valid, using macro variable for max CPE's with a default value
option 18 ${MAX_CPES, 1}


# Valid, using macro variable for vendor option
option 43.200 hex ${MACRO_VAR_HEX}

# Valid, using macro variable for vendor option
option 43.201 ascii ${MACRO_VAR_ASCII}

# Valid, using macro variable for vendor option
option 43.202 ip ${MACRO_VAR_IP}

# Valid, using macro variable in double quotes
option 18 "${MAX_CPES}"

# Valid, using macro variable within a value
option 43.131 ascii "hostname ${HOSTNAME}"

# Valid, using macro variables in multi-valued options
option 11 ${ACCESS_CONTROL_MIB,
.mib-2.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccessEntry.docsDevNmAc
cessControl.1}, Integer, ${ACCESS_CONTROL_VAL, 3}

# Valid, using macro variable in an include statement
include "${EXTRA_TEMPLATE}"

# Valid, using macro variable in an include statement with a default value
include "${EXTRA_TEMPLATE, modem_reset.tmpl}"

# Valid, using macro variable in an include statement with a default value
include "${EXTRA_TEMPLATE, modem_reset}.tmpl"

# Valid, using macro variable in an include statement with an ignore clause
include "${MY_TEMPLATE, ignore}"
```

例 5-14　マクロ変数の誤った使用方法

```
# Invalid, using macro variable as the option number
option ${MAX_CPES} 1

# Invalid, using macro variable with space in name
option 18 ${MAX CPES}
```

SNMP TLV

BAC は、Option 11 および 64 を使用して、動的テンプレート ファイルの SNMP TLV をサポートします。次のものが対象です。

- DOCSIS：Broadband Access Center for Cable（BACC）バージョン 2.0 以降。
- PacketCable：BACC バージョン 2.5 以降。
- CableHome：BACC バージョン 2.6 以降。

これらのテンプレート ファイルの SNMP TLV の構文を検証する場合、BAC は SNMP TLV で参照される対応する SNMP OID を含む MIB ファイルを必要とします。テンプレートに、MIB で検出できない SNMP OID を備えた SNMP TLV が含まれている場合、SNMP TLV は構文エラーを生成します。

次の各項では、MIB を使用しない SNMP TLV、またはベンダー固有の MIB を使用した SNMP TLV の追加方法について説明します。

MIB を使用しない SNMP TLV の追加

RDU で MIB をロードしなくても、動的設定ファイル（DOCSIS、PacketCable、CableHome）に SNMP TLV を追加できます。次の方法を使用すると、RDU 設定の拡張から、DOCSISOptionFactory インターフェイスを使用して機能にアクセスできます。

```
public OptionValue createOptionValue(OptionSyntax syntax, String optionNumStr,
String[] optionValueList)
```

上記の方法では、public OptionSyntax.SNMP 列挙値を、OID、Type、Value という値のセットを含む optionValueList と組み合せて使用できます。

RDU 動的設定テンプレートから、次の構文を使用して、RDU MIB に対して検証されていない SNMP TLV を指定します。

```
option option-number snmp OID, Type, Value
```

例：

```
# DOCS-CABLE-DEVICE-MIB:
option 11 snmp .docsDevNmAccessIp.1,IPADDRESS,192.168.1.1

# Arris vendor specific SNMP TLV (OID numbers only, mix names/numbers)
option 11 snmp .1.3.6.1.4.1.4115.1.3.1.1.2.3.2.0, INTEGER, 6
option 11 snmp .enterprises.4115.1.3.1.1.2.3.2.0, INTEGER, 6

# NOTE: trailing colon required for single octet
option 11 snmp .1.3.6.1.2.1.69.1.2.1.6.3, STRING, 'c0:'
```

表 5-2 は、利用可能な SNMP 変数タイプの名前です。

表 5-2　SNMP 変数タイプ

IETF 標準の SMI データ タイプ	SNMP API の名前
Integer32	INTEGER
Integer（Enumerated）	INTEGER
Unsigned32	UNSIGNED32
Gauge32	GAUGE
Counter32	COUNTER
Counter64	COUNTER64

表 5-2　SNMP 変数タイプ（続き）

IETF 標準の SMI データ タイプ	SNMP API の名前
Timeticks	TIMETICKS
OCTET STRING	STRING
OBJECT IDENTIFIER	OBJID
IpAddress	IPADDRESS
BITS	STRING

たとえば、SMI Integer32 タイプを指定する場合は、Integer32 および INTEGER タイプが利用可能です（大文字と小文字は区別されません）。

OCTET STRING タイプの場合は、OCTET STRING、OCTETSTRING、または STRING タイプがすべて利用可能です。

カスタム SNMP TLV テンプレート オプションを使用して、任意の SNMP TLV（RDU MIB に存在するものを含む）を指定することができます。カスタム SNMP TLV エラー チェックはあまり厳しくないため、誤ったスカラー / テーブルの参照は検出されません（たとえば、OID 名における .0 と .n の区別）。

ベンダー固有の MIB を使用した SNMP TLV の追加

MIB を RDU に追加すると、テンプレートで人間が判読可能な SNMP OID を使用できるようになり、さらに、マクロ変数に SNMP TLV 値を使用できるようになります。

BACC 2.6 以前

使用する SNMP OID に対応する MIB を保持している場合、MIB ファイルを BAC RDU に追加できます。MIB を追加後、新しい MIB で参照される SNMP OID を使用する SNMP TLV が認識されます。

新しい MIB を RDU に追加するには、次の手順に従います。

ステップ 1　新しい MIB ファイルを *BPR_HOME/rdu/mibs* ディレクトリにコピーします。

ステップ 2　*/docsis/mibs/custom/mibList* プロパティを次の場所にコピーします。このプロパティ値には、MIB ファイル名のカンマ区切りリストが含まれます。

a. *rdu.properties* ファイル。このファイルは、RDU と管理者のユーザ インターフェイスが使用します。このファイルは *BPR_HOME/rdu/conf* ディレクトリにあります。

b. *api.properties* ファイル。このファイルは設定ファイル ユーティリティ（runCfgUtil.sh ツール）が使用します。

（注） *api.properties* ファイルは、BAC のインストール処理では作成されません。このファイルは、初めて使用するときに任意のテキスト エディタを使用して手動で作成する必要があります。このファイルは *BPR_HOME/rdu/conf* ディレクトリに置いてください。

api.properties ファイルには、*/docsis/mibs/custom/mibList* が含まれます。これは、Arris embedded MTA（eMTA）で使用できる MIB のセット用に設定されます。

ステップ 3　**/etc/init.d/bprAgent restart rdu** コマンドを使用して、BAC プロセス ウォッチドッグ経由で RDU を再起動します。

次の例では、ARRIS MTA を設定するためにテンプレートで使用する追加の ARRIS MIB について説明しています。

Arris ベンダー固有の SNMP TLV 使用すると仮定します。

```
option 11 .ppCfgMtaCountryTemplate.0, INTEGER, 9
```

次の MIB ファイルが利用可能になります。

- ARRIS-MIB
- ARRIS-CM-CAPABILITY-MIB
- ARRIS-CM-DEVICE-MIB
- ARRIS-MTA-DEVICE-MIB
- PACKETPORT-MIB

MIB ファイルを *BPR_HOME/rdu/mibs* ディレクトリにコピーし、次のプロパティを *api.properties* ファイルと *rdu.properties* ファイルに挿入する必要があります。

/docsis/mibs/custom/mibList=ARRIS-MIB,ARRIS-CM-CAPABILITY-MIB,ARRIS-CM-DEVICE-MIB,ARRIS-MTA-DEVICE-MIB,PACKETPORT-MIB

BACC 2.7 以降

（注） BACC 2.7 以降では、*/docsis/mibs/custom/mibList* プロパティは、*/snmp/mibs/mibList* に名前が変更されています。

使用する SNMP OID に対応する MIB を保持している場合、MIB ファイルを BAC RDU に追加できます。MIB を追加後、新しい MIB で参照される SNMP OID を使用する SNMP TLV が認識されます。

新しい MIB を BAC RDU に追加するには、次の手順に従います。

ステップ 1　BAC 管理者のユーザ インターフェイスを起動します。

ステップ 2　ナビゲーション バーで、**Configuration** > **Defaults** をクリックします。

ステップ 3　表示される Configure Defaults ページの左ペインにある System Defaults リンクをクリックします。

ステップ 4　MIB List フィールドの末尾に新しい MIB の内容を貼り付けます。

ステップ 5　**Submit** をクリックします。

（注） バージョン 2.7 以降では、MIB の解析ツールは強化されています。その結果、以前はエラーなしで解析されていた MIB のバージョンで時々エラーが返されるようになりました。エラーが発生し、新しい MIB を編集することで解決できない場合は、Cisco Technical Assistance Center にお問い合せください。

MIB のロード順のデバッグ

一般に、ベンダーが提供するさまざまな MIB は、MIB 間の依存関係を満たすために特定の順番でロードする必要があります。ただし、多くの場合、ベンダーは正しいロード順を提供しないので、ユーザ自身が正しいロード順を決定する必要があります。この項では、BAC のデバッグ情報を使用して、MIB のロード順の問題を解決する方法について説明します。

（注） BAC 内の MIB のロード順は、次のプロパティにリストされている MIB の順番で設定されます。

- */docsis/mibs/custom/mibList* プロパティ（BACC 2.6.x 以前のリリースを使用している場合）。
- */snmp/mibs/MibList* プロパティ（BACC 2.7.x 以降のリリースを使用している場合）。

runCfgUtil.sh ツールを使用して、*api.properties* ファイルで指定されているプロパティの正しいロード順を判断できます。runCfgUtil.sh ツールは、*BPR_HOME/rdu/bin* ディレクトリにあります。

（注） この手順では、BACC 2.7.x 以降のリリースで使用する */snmp/mibs/MibList* プロパティを参照します。2.6.x 以前のリリースを実行している場合は、*/docsis/mibs/custom/mibList* プロパティを使用してください。

ステップ 1　*api.properties* ファイルを使用して Configure runCfgUtil.sh を設定し、このステップで説明されているのと同様の設定内容を使用します。*api.properties* ファイルは、BAC トレースをイネーブルにし、MIB デバッグ情報をユーザ コンソールに送信します。

```
#
# Enable logging to the console
#
/server/log/1/level=Info
/server/log/1/properties=level
/server/log/1/service=com.cisco.csrc.logging.SystemLogService
/server/log/1/name=Console
#
# Enable trace categories
#
/server/log/trace/rduserver/enable=enabled
#
# The list of MIBs to be added.
#
/snmp/mibs/MibList=arrishdr.mib,arris_cm_capability.mib,arris_mta_device.mib,arris_sip
.mib,arris_cm.mib,pp.mib,blp2.mib,dev0.mib,docs_evnt.mib,qos.mib,test.mib,usb.mib,snmp
v2_conf.mib,rfc1493.mib,rfc1907.mib,rfc2011.mib,rfc2013.mib,rfc2233.mib,rfc2571.mib,rf
c2572.mib,rfc2573.mib,rfc2574.mib,rfc2575.mib,rfc2576.mib,rfc2665.mib,rfc2669.mib,rfc2
670.mib,rfc2786.mib,rfc2851.mib,rfc2933.mib,rfc 3083.mib
```

ステップ 2　runCfgUtil.sh をこのように設定し、ツールを実行して、Option 11 または Option 64（SNMP 符号化）を含んでいる任意のテンプレートを符号化します。ツールは、`/snmp/mibs/MibList` 内で指定されている MIB のロードを試み、MIB のロード エラーとともに完全なデバッグ情報をユーザ コンソールに送信します。

ステップ 3　エラー情報を使用して、MIB の全セットがエラーなしでロードされ、ファイルの符号化が成功するまで、`/snmp/mibs/MibList` 内で指定されている MIB の順番を変更します。

ステップ 4　正常なロード順を決定したら、使用している BACC のバージョンに基づき、このステップで説明されている手順を実行します。

BACC 2.7 以降

a. 管理者のユーザ インターフェイスから、**Configuration > Defaults** をクリックして、System Defaults リンクをクリックします。

b. MIB List フィールドに、ロード順の情報をコピーします。

これで、RDU はベンダー提供の MIB を使用してテンプレートを符号化するように設定されます。

（注） RDU を再起動する必要はありません。

api.properties ファイルと MIB List フィールドでは、*/snmp/mibs/mibList* 文字列を使用してください。

BACC 2.6 以前

a. *rdu.properties* ファイルの */docsis/mibs/custom/mibList* プロパティにロード順の情報をコピーします。このファイルは *BPR_HOME/rdu/conf* ディレクトリにあります。

b. **/etc/init.d/bprAgent restart rdu** コマンドを使用して、BAC プロセス ウォッチドッグ経由で RDU を再起動します。

これで、RDU はベンダー提供の MIB を使用してテンプレートを符号化するように設定されます。

定義済みオプションの符号化タイプ

表 5-3 に、定義済み符号化タイプを持つオプションを示します。

表 5-3　定義済みオプション符号化タイプ

符号化	入力	例
認可アクション	8 ビット符号なし整数または説明のための文字列。 認可を許可する場合の値は次のとおりです。 • 0 • permit 認可を拒否する場合の値は次のとおりです。 • 1 • deny	0 1 permit deny

表 5-3　定義済みオプション符号化タイプ（続き）

符号化	入力	例
アクセス ビュー コントロール	8 ビット符号なし整数または説明のための文字列。 アクセス ビューの SNMPv3 Access View サブツリーを含める場合の値は次のとおりです。 • 1 • included アクセス ビューの SNMPv3 Access View サブツリーを除外する場合の値は次のとおりです。 • 2 • excluded	1 2 included excluded
アクセス ビュー タイプ	8 ビット符号なし整数または説明のための文字列。 読み取り専用アクセスをイネーブルにする場合の値は次のとおりです。 • 1 • read-only 読み取りと書き込みアクセスをイネーブルにする場合の値は次のとおりです。 • 2 • read-write	1 2 Read-only Read-write
ActInact	8 ビット符号なし整数または説明のための文字列。 TLV をディセーブルにする場合の値は次のとおりです。 • 0 • inactive TLV をイネーブルにする場合の値は次のとおりです。 • 0 • active	0 1 Inactive Active
BitFlag8	8 ビット符号なし整数。出力は、値を表す 16 進数の文字列です。	0xFE
BitFlag32	32 ビット符号なし整数。出力は、値を表す 16 進数の文字列です。	0xFFFF0000
ブール値	0 は false、1 は true です。	0 1

表 5-3　定義済みオプション符号化タイプ（続き）

符号化	入力	例
Byte16	32 文字の 16 進数文字列で指定される 16 バイト。これは通常、ケーブル モデムと CMTS の MIC オプションを表すために使用されます。0x プレフィックスは使用できません。	なし。 BAC はケーブル モデムと CMTS MIC オプションのハッシュを自動的に計算します。
バイト	一連の 16 進数オクテット。各オクテットは 2 文字にする必要があります。	`000102030405060708`

CPE アクセス コントロール	8 ビット符号なし整数または説明のための文字列。
	デバイス アクセス コントロールをディセーブルにする場合の値は次のとおりです。
	• 0
	• ディセーブル
	デバイス アクセス コントロールをイネーブルにする場合の値は次のとおりです。
	• 1
	• イネーブル
		`0`
		`1`
		`Disabled`
		`Enabled`

DSC 分類子	8 ビット符号なし整数または DSC 分類子の文字列名。符号なし整数には次のものがあります。 • 0：DSC 追加分類子 • 1：DSC 置換分類子 • 2：DSC 削除分類子	`0`

EnableDisable	8 ビット符号なし整数または説明のための文字列。
	ディゼーブルにする場合の値は次のとおりです。
	• 0
	• ディセーブル
	イネーブルにする場合の値は次のとおりです。
	• 1
	• イネーブル
		`0`
		`1`
		`Disabled`
		`Enabled`

表 5-3　定義済みオプション符号化タイプ（続き）

符号化	入力	例
Inet アドレス ピア	1 バイトの InetAddressTypeCode。 • 1 は IPv4 • 2 は IPv6 この値の後に IPv4 または IPv6 インターネット アドレスが続きます。 その結果、この長さは IPv4 用が 5 バイト（1+4）、IPv6 用が 17 バイト（1+16）になります。	1,10.112.125.111 2,0:0:0:0:0:ffff:8190:3426
IP アドレス	ドット（.）で区切られた 4 つの符号なし整数 8。	10.10.10.1
IPv6 アドレス	IPv6 アドレス *x:x:x:x:x:x:x:x* を表す文字列。ここで、*x* は、アドレスの 8 つの 16 ビット部分を表す 1 ～ 4 桁の 16 進数です。	2001:db8:0:0:8:800:200c:417a
IPv4 または IPv6 アドレス	IPv4 または IPv6 アドレスを表す文字列。	10.112.125.111 0:0:0:0:0:ffff:8190:3426
複数の IP アドレス	IP アドレスのカンマ区切りリスト。	10.11.12.13,10.11.12.14
複数の IPv6 アドレス	IPv6 アドレスのカンマ区切りリスト。	2001:db8:0:0:8:800:200c:417a,ff01:0:0:0:0:0:0:101
MAC アドレス	コロン（:）またはダッシュ（-）で区切られた 6 つの 16 進数オクテット。各オクテットは正確に 2 文字にする必要があります。コロンとダッシュを同時に使用することはできません。	00:01:02:03:04:05 00-01-02-03-04-05
MAC アドレスとマスク	コロン（:）またはダッシュ（-）で区切られた 12 のオクテット。各オクテットは 2 文字にする必要があります。コロンとダッシュを同時に使用することはできません。最初の 6 つのオクテットは MAC アドレスを表し、残りの 6 つは MAC アドレスのマスクを表します。	00:01:02:03:04:05:06:07:08:09:0A:0B 00-01-02-03-04-05-06-07-08-09-0A-0B
NoLV	タイプのみ。値や長さは含みません。	null
NVTASCII	ASCII 文字列。符号化文字列を NULL で終わることはできません。	This is an ASCII string
OID	SNMP OID 文字列。	sysinfo.0
OIDCF	SNMP OID 文字列とカンマで区切られた符号なし整数（0 または 1）。	sysinfo.0,1

表 5-3　定義済みオプション符号化タイプ（続き）

符号化	入力	例
OnOff	8 ビット符号なし整数。 TLV をオンにする場合の値は次のとおりです。 • 0 • On TLV をオフする場合の値は次のとおりです。 • 1 • Off	0 1 On Off
OUI	コロン（:）またはダッシュ（-）で区切られた 3 つの 16 進数オクテット。各オクテットは 2 文字にする必要があります。	00-00-0C
RFC868 時刻	RFC868 時刻を表す 32 ビット符号なし整数。出力は、*MM/dd/yyyy HH:mm:ss* の形式の日付と時刻の文字列です。	0 （"12/31/1899 19:00:00" を表します） 4294967295 （"02/07/2036 01:28:15" を表します）
サービス フロー	8 ビット符号なし整数またはサービス フローの説明のための文字列。出力は次の内容を示すサービス フローです。 • 0：予約済み • 1：未定義（CMTS の実装に依存） • 2：ベスト エフォート • 3：非リアルタイム ポーリング サービス • 4：リアルタイム ポーリング サービス • 5：任意の認可サービス アクティビティ検出 • 6：任意の認可サービス	0

表 5-3　定義済みオプション符号化タイプ（続き）

符号化	入力	例
SNMPVarBind	SNMP OID 文字列、タイプ、値。これらはそれぞれカンマで区切られます。有効な値は次のとおりです。 • BITS • Counter • Counter32 • Counter64 • Gauge • Gauge32 • INTEGER • Integer32 • IpAddress • OCTETSTRING • OBJECTIDENTIFIER • Opaque • TimeTicks • Unsigned32 （注）OCTETSTRING は、末尾に NULL を含まない 16 進表記に変換される文字列（オクテット文字列など）または引用符で囲まれた 16 進表記（'aa:bb:cc' など) です。	`.experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccessEntry.docsDevNmAccessStatus.1, INTEGER, 4`
SrvChangeAct	0 ～ 3 の範囲に限定された 8 ビット符号なし整数または SrvChangeAct の説明。説明のための文字列の出力は次のとおりです。 • 0：PHS ルールの追加 • 1：PHS ルールの設定 • 2：PHS ルールの削除 • 3：すべての PHS ルールの削除	`0`
サブタイプ	1 つまたは 2 つのカンマで区切られた符号なし整数 8。	`12` `12,14`

表 5-3　定義済みオプション符号化タイプ（続き）

符号化	入力	例

転送アドレスとマスク	IPv4 の場合、ドット付き表記の 4 つのオクテット IP アドレスと、カンマ（,）で区切られたポート番号。
	IPv6 の場合、次のようなドット付き表記または文字列です。
	• ドット付き表記の有効な IPv6 アドレスと、カンマ（,）で区切られたポート番号。
	• IPv6 アドレスを表す文字列と、カンマ（,）で区切られたポート番号。たとえば、*x:x:x:x:x:x:x:x*,1234 の場合、*x* は、アドレスの 8 つの 16 ビット部分を表す 1 ～ 4 桁の 16 進数です。
		IPv4
		`10.112.125.111,5678`
		IPv6
		`2001.db8.0.0.8.800.200c.417a,5678`
		`2001:db8:0:0:8:800:200c:417a,5678`

8 ビット符号なし整数	0 ～ 255	`14`
16 ビット符号なし整数	0 ～ 65535	`1244`
32 ビット符号なし整数	0 ～ 4294967295	`3455335`
8 ビット符号なし整数と 16 ビット符号なし整数	カンマで区切られた 1 つの符号なし整数 8 と 1 つの符号なし整数 16。	`3,12324`
8 ビット符号なし整数のペア	カンマで区切られた 2 つの符号なし整数 8。	`1,3`
3 ビット バイトの 8 ビット符号なし整数	カンマで区切られた 3 つの符号なし整数 8。	`1,2,3`
検証	8 ビット符号なし整数 検証をイネーブルにする場合の値は次のとおりです。 • 0 • 検証 検証をディゼーブルにする場合の値は次のとおりです。 • 1 • Don't Verify **（注）** Verify TLV の true と false の定義は、DOCSIS 1.1 の仕様（Option 26.11）と一致しています。	`0 = verify` `1 = don’t verify`
ZTASCII	ASCII 文字列。符号化文字列は NULL で終わります。	`This is an ASCII string`

BITS 値の構文

BITS 型を使用する場合は、ラベル（「interval1 interval2 interval3」）または数値による位置（「0 1 2」）を指定する必要があります。ラベル値は 1 ベースで、ビット値は 0 ベースであることに注意してください。

ビット番号を使用する構文の例を示します。

```
option 11 .pktcSigDevR0Cadence.0,STRING,"0 1 2 3 4 5 6 7 8 9 10 11 12 13 14"
```

ラベルを使用するカスタマー オクテット文字列（FFFE000000000000）の構文を示します。

```
option 11 .pktcSigDevR0Cadence.0,STRING,"interval1 interval2 interval3
interval4 interval5 interval6 interval7 interval8 interval9 interval10
interval11 interval12 interval13 interval14 interval15"
```

OCTETSTRING の構文

OCTETSTRING は、末尾に NULL を含まない 16 進表記に変換される文字列（オクテット文字列など）または一重引用符で囲まれた 16 進表記（'aa:bb:cc' など）です。

設定ファイル ユーティリティの使用方法

設定ファイル ユーティリティを使用して、PacketCable 1.0/1.1/1.5、DOCSIS 1.0/1.1/2.0/3.0、および CableHome のテンプレート ファイルと設定ファイルをテスト、検証、および表示できます。これらの作業は、独自の設定ファイルを正常に展開するために重要です。テンプレートの詳細については、P.5-2 の「テンプレート ファイル：概要」を参照してください。

設定ファイル ユーティリティは、RDU をインストールしたときにのみ利用可能です。このユーティリティは *BPR_HOME/rdu/bin* ディレクトリにインストールされます。

符号化するテンプレート ファイルとデコードするバイナリ ファイルの両方が、設定ファイル ユーティリティを起動するディレクトリに存在する必要があります。

この項のすべての例では、RDU が運用中で、次の条件が適用されていることを前提にしています。

- BAC アプリケーションは、デフォルトのホームディレクトリ（*/opt/CSCObac*）にインストールされています。
- RDU ログイン名は **admin** です。
- RDU ログイン パスワードは **changeme** です。

（注） この項の例では、一部が出力例にとって重要でない場合に、その部分を省略していることがあります。その場合は、例中のサマリーの直前に省略記号（...）を示しています。

この項では、次のトピックについて取り上げます。

設定ファイル ユーティリティの実行

次の手順と例で、「設定ファイル ユーティリティを実行する」というフレーズは、指定されたディレクトリから **runCfgUtil.sh** コマンドを入力することを意味します。設定ファイル ユーティリティを実行するには、*BPR_HOME/rdu/bin* ディレクトリから次のコマンドを実行します。

runCfgUtil.sh *options*

利用可能な *options* は、次のとおりです。

- ***-c*** *secret*：DOCSIS テンプレート ファイルを解析するときの CMTS 共有秘密情報を指定します。デフォルトの共有秘密情報を指定するには、**-c cisco** と入力します。
- ***-cablehome***：入力ファイルが CableHome ポータル サービス設定ファイルであることを示します。***-docsis*** または ***-pkt*** オプションと同時に使用することはできません。
- ***-d***：バイナリ入力ファイルをデコードします。**-e** オプションと同時に使用することはできません。
- ***-docsis***：入力ファイルが DOCSIS 設定ファイルであることを指定します。このデフォルトを ***-pkt*** オプションと同時に使用することはできません。
- **-v** *version*：使用している DOCSIS のバージョンを指定します。たとえば、DOCSIS 1.1 を使用している場合は、**-v 1.1** と入力します。バージョン番号を指定しない場合、コマンドはデフォルトで DOCSIS 2.0 を使用します。BAC がサポートする値は 1.0、1.1、2.0、および 3.0 です。
- ***-e***：テンプレート入力ファイルを符号化します。このデフォルトを **-d** オプションと同時に使用することはできません。
- ***-g***：DOCSIS、PacketCable、または CableHome バイナリ ファイルからテンプレート ファイルを生成します。
- ***-h*** *host:port*：ホストとポート番号を指定します。デフォルトのポート番号は 49187 です。
- ***-i*** *device-id*：テンプレート解析中にマクロ変数を代入するときに使用するデバイスを示します。たとえば、デバイス MAC アドレスが 1,6,00:00:00:00:00:00:01 の場合は、**-i 1,6,00:00:00:00:00:00:01** と入力し、デバイス DUID が 00:03:00:01:00:18:68:52:75:c0 の場合は、**-i 00:03:00:01:00:18:68:52:75:c0** と入力します。このオプションを使用するときは、**-u** および **-p** オプションを使用して、それぞれユーザ名とパスワードを指定する必要もあります。**-m** オプションと同時に使用することはできません。
- ***-l*** *filename*：入力ファイルがローカル ファイル システムにあることを示します。たとえば、入力ファイルの名前が *any_file* の場合、**-l any_file** と入力します。**-r** オプションと同時に使用することはできません。
- ***-loc***：PacketCable のロケールを **na**（北アメリカ）または **euro**（ヨーロッパ）に指定します。デフォルトは na です。MTA が euro-MTA の場合は、ロケールを euro に設定する必要があります。
- ***-m*** *macros*：マクロ変数のキーと値のペアを指定します。形式は key=value です。複数のマクロ変数が必要な場合は、key_1=value_1,,key_2=value_2 のように、キーと値のペアを 2 つのカンマで区切ります。**-i** オプションと同時に使用することはできません。
- ***-p*** *password*：RDU に接続するときに使用するパスワードを指定します。たとえば、パスワードが 123456 の場合は、***-p 123456*** と入力します。
- ***-o*** *filename*：解析したテンプレート ファイルをバイナリ ファイルとして保存します。たとえば、出力を *op_file* という名前のファイルに保存するには、**-o op_file** と入力します。
- ***-pkt***：入力ファイルが PacketCable MTA 設定ファイルであることを示します。***-docsis*** オプションと同時に使用することはできません。
- ***-r*** *filename*：入力ファイルが RDU に追加したリモート ファイルであることを示します。たとえば、ファイル名が *file25* の場合は、**-r file25** と入力します。このオプションを使用するときは、**-u** および **-p** オプションを使用して、それぞれユーザ名とパスワードを指定する必要もあります。**-l** オプションと同時に使用することはできません。
- **-s**：解析されたテンプレートまたはバイナリ ファイルの内容を人間が判読可能な形式で表示します。
- **-t**：PacketCable 符号化タイプを **Secure** または **Basic** に指定します（デフォルトは Secure）。

- **-u** *username*：RDU に接続するときに使用するユーザ名を指定します。たとえば、ユーザ名が admin の場合、**-u admin** と入力します。

（注） 設定ファイル ユーティリティでは、テンプレート ファイルに Option 19（TFTP サーバ タイムスタンプ）と Option 20（TFTP サーバのプロビジョニングされたモデム アドレス）は含まれません。ただし、BAC TFTP 混在では含まれます。また、options 6（CM MIC）および 7（CMTS MIC）はどちらも、符号化されたテンプレート ファイルに自動的に挿入されます。そのため、これらの Message Integrity Check（MIC; メッセージ完全性チェック）を指定する必要はありません。

BAC へのテンプレートの追加

設定ファイル ユーティリティを使用して、BAC テンプレートをテストするには、次の手順に従います。

ステップ 1　P.5-2 の「テンプレート ファイル：概要」の説明に従い、テンプレートを作成します。テンプレートに他のテンプレートを含める場合は、参照されるテンプレートすべてが同一のディレクトリにあることを確認します。

ステップ 2　ローカル ファイル システムで設定ファイル ユーティリティを実行します。テンプレートの構文をチェックするか、または CRS と同じ方法で設定ファイル ユーティリティにテンプレートを処理させた後、出力を返すことができます。

テンプレートにマクロ変数が含まれる場合、指定された順番で次の操作を実行します。

a.　コマンドラインの代入を使用してテストします。

b.　RDU に追加したデバイスを使用してテストします。

ステップ 3　テンプレート（および、そのテンプレートにインクルードするテンプレート）を RDU に追加します。

ステップ 4　設定ファイル ユーティリティを実行して、ファイルを解析します。P.5-29 の「外部テンプレート ファイルのテンプレート処理のテスト」を参照してください。

テンプレートにマクロ変数が含まれる場合、指定された順番で次の操作を実行します。

a.　コマンドラインの代入を使用してテストします。

b.　RDU に追加したデバイスを使用してテストします。

ステップ 5　テストが成功したら、そのテンプレートを使用するサービス クラスを設定します。

テンプレート ファイルへのバイナリ ファイルの変換

runCfgUtil.sh コマンドを使用して、バイナリ設定メモリ ファイルをテンプレート ファイルに変換します。BAC の動的構成生成は、作成されるテンプレートに基づきます。既存のテスト済みバイナリ ファイルをテンプレート ファイルに自動的に変換すると、プロセスの速度が向上し、エラーが発生する可能性は低下します。

シンタックスの説明　***runCfgUtil.sh -g -l*** *binary_file* ***-o*** *template_file*

- ***-g***：入力バイナリ ファイルからテンプレート ファイルを生成する必要があることを指定します。
- ***-l*** *binary_file*：ローカル入力ファイル（パス名を含む）を指定します。すべての場合において、入力バイナリ ファイルの名前には *.cm* ファイル拡張子が割り当てられます（たとえば、*bronze.cm*）。
- ***-o*** *template_file*：出力テンプレート ファイル（パス名を含む）を指定します。すべての場合において、出力テンプレート ファイルの名前には *.tmpl* ファイル拡張子が割り当てられます（たとえば、*test.tmpl*）。

バイナリ ファイルをテンプレート ファイルに変換するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/rdu/samples/docsis* にディレクトリを変更します。

ステップ 2　使用するテンプレート ファイルを選択します。この例では、既存のバイナリ ファイル *unprov.cm* を使用します。

ステップ 3　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -g -l unprov.cm -o test.tmpl -docsis
```

-docsis：入力ファイルを DOCSIS 設定ファイルにすることを指定します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26

################################################################
## Template File Generator
## Generated on Fri Oct 12 16:12:51 EST 2007
################################################################

################################################################
## Each generated option will be represented by the following:
## The first line will represent a description of the
## generated option
## The second line will represent the generated option
## The third line will represent the custom version
## of the generated option
################################################################

# (3) Network Access Control
Option 3  01
# Option 3  hex 01

# (4.1) Class ID
Option 4.1 1
# Option 4.1 hex 01

# (4.2) Maximum Downstream Rate
Option 4.2 128000
# Option 4.2 hex 0001F400

# (4.3) Maximum Upstream Rate
Option 4.3 64000
# Option 4.3 hex 0000FA00

# (4.4) Upstream Channel Priority
Option 4.4 1
# Option 4.4 hex 01

# (4.5) Guaranteed Minimum Upstream Channel Data Rate
Option 4.5 0
# Option 4.5 hex 00000000

# (4.6) Maximum Upstream Channel Transmit Burst
Option 4.6 1600
# Option 4.6 hex 0640

# (4.7) Class-of-Service Privacy Enable
Option 4.7 00
# Option 4.7 hex 00

# (11) SNMP MIB Object
Option 11
.iso.org.dod.internet.experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docs
DevNmAccessEntry.docsDevNmAccessStatus.1,INTEGER,createAndGo
# Option 11  hex 3082000F060A2B06010353010201070102010 4

...

# (18) Maximum Number of CPEs
Option 18  1
# Option 18  hex 01
```

ローカル テンプレート ファイルのテンプレート処理のテスト

runCfgUtil.sh コマンドを使用して、ローカル ファイル システムに格納されているテンプレート ファイルの処理をテストします。

シンタックスの説明　***runCfgUtil.sh -pkt -l*** *file*

- ***-pkt***：入力ファイルが PacketCable MTA ファイルであることを示します。
- ***-l***：入力ファイルがローカル ファイル システムにあることを指定します。
- *file*：解析する入力テンプレート ファイルを示します。

ローカル ファイル システムにあるテンプレート ファイルを解析するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/rdu/samples/packet_cable* にディレクトリを変更します。

ステップ 2　使用するテンプレート ファイルを選択します。この例では、既存のテンプレート ファイル *unprov_packet_cable.tmpl* を使用します。これは PacketCable MTA テンプレートであるため、**-pkt** オプションを使用します。

ステップ 3　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -pkt -l unprov_packet_cable.tmpl
```

unprov_packet_cable.tmpl：解析する入力テンプレート ファイルを示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes      Option    Description              Value
0       FE0101          254       Telephony Config File    1
                                  Start/End
3       0B153013060E    11        SNMP MIB Object          .iso.org.dod.internet.privat
        2B06010401A30B                                     e.enterprises.cableLabs.clab
        0202010101                                         Project.clabProjPacketCable.
        0700020102                                         pktcMtaMib.pktcMtaMibObjects
                                                           .pktcMtaDevBase.
                                                           pktcMtaDevEnabled.0,INTEGER,
                                                           false(2)
...

0 error(s), 0 warning(s) detected. Parsing of unprov_packet_cable.tmpl was successful.
The file unprov_packet_cable.tmpl was parsed successfully in 434 ms.
The parser initialization time was 92 ms.
The parser parse time was 342 ms.
```

外部テンプレート ファイルのテンプレート処理のテスト

runCfgUtil.sh コマンドを使用して、外部テンプレート ファイルの処理をテストします。

シンタックスの説明　***runCfgUtil.sh -docsis -r*** *file* ***-u*** *username* ***-p*** *password*

- ***-r***：入力ファイルが RDU に追加したファイルであることを示します。
- *file*：解析する入力テンプレート ファイルを示します。
- ***-u*** *username*：RDU に接続するときに使用するユーザ名を指定します。
- ***-p*** *password*：RDU に接続するときに使用するパスワードを指定します。
- **-docsis**：ファイルが DOCSIS テンプレートであることを示します。

RDU に追加したテンプレート ファイルを解析するには、次の手順に従います。

ステップ 1　使用するテンプレート ファイルを選択します。この例では、既存のテンプレート ファイル *unprov.tmpl* を使用します。DOCSIS テンプレートを使用するため、**-docsis** オプションを使用します。

ステップ 2　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -docsis -r unprov.tmpl -u admin -p changeme
```

- **unprov.tmpl**：入力ファイルを示します。
- **admin**：ユーザ名を示します。
- **changeme**：パスワードを示します。

ユーティリティを実行すると、次のような結果が表示されます。

（注）　ここに表示されている結果は説明のためだけのものであり、簡潔にするために省略されています。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes       Option    Description                        Value
0       030101           3         Network Access Control             On
3       041F             4         Class of Service
5       010101           4.1       Class ID                           1
8       02040000FA00     4.2       Maximum Downstream Rate            128000 bits/sec
14      03040000FA00     4.3       Maximum Upstream Rate              64000 bits/sec
20      040101           4.4       Upstream Channel Priority          1
...
252     06108506547F     6         CM MIC Configuration Setting       8506547FC9152B44
        C9152B44DB95                                                  DB955420843EF6FE
        5420843EF6FE
270     0710644B675B     7         CMTS MIC Configuration Setting     644B675B70B7BD3E
        70B7BD3E09AC                                                  09AC210F794A1E8F
        210F794A1E8F
288     FF               255       End-of-Data Marker
289     00               0         PAD
290     00               0         PAD
291     00               0         PAD
0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.
The file unprov.tmpl was parsed successfully in 375 ms.
The parser initialization time was 63 ms.
The parser parse time was 312 ms.
```

ローカル テンプレート ファイルのテンプレート処理のテストと共有秘密情報の追加

runCfgUtil.sh コマンドを使用して、テンプレート ファイルの処理をテストし、指定する共有秘密情報を追加します。

シンタックスの説明　***runCfgUtil.sh -e -docsis -l*** *file* ***-c*** *secret*

- ***-e***：符号化オプションを示します。
- ***-docsis***：入力ファイルが DOCSIS テンプレート ファイルであることを示します。
- *-l*：入力ファイルがローカル ファイル システムにあることを指定します。
- *file*：解析する入力テンプレート ファイルを示します。
- ***-c***：DOCSIS テンプレート ファイルを解析するときの CMTS 共有秘密情報を指定します。
- *secret*：新しい共有秘密情報を示します。デフォルトの共有秘密情報は **cisco** です。

ローカルに保存したテンプレート ファイルを解析し、ユーザ固有の共有秘密情報を設定するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/rdu/templates* にディレクトリを変更します。

ステップ 2　解析するテンプレート ファイルを選択します。この例では、既存のテンプレート ファイル *unprov.tmpl* を使用します。これは DOCSIS テンプレートであるため、**-docsis** オプションを使用します。

ステップ 3　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -e -docsis -l unprov.tmpl -c shared
```

- **unprov.tmpl**：ローカル ファイル システムの入力ファイルを示します。
- **shared**：新しい共有秘密情報を示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes        Option     Description                     Value
0       030100            3         Network Access Control           Off
3       041F              4         Class of Service
5       010101            4.1       Class ID                         1
8       02040001F400      4.2       Maximum Downstream Rate          128000 bits/sec
14      03040000FA00      4.3       Maximum Upstream Rate            64000 bits/sec
20      040101            4.4       Upstream Channel Priority        1
...
252     06108506547F      6         CM MIC Configuration Setting     8506547FC9152B44
        C9152B44DB95                                                 DB955420843EF6FE
        5420843EF6FE
270     0710644B675B      7         CMTS MIC Configuration Setting   644B675B70B7BD3E
        70B7BD3E09AC                                                 09AC210F794A1E8F
        210F794A1E8F
288     FF                255       End-of-Data Marker
289     00                0         PAD
290     00                0         PAD
291     00                0         PAD
0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.
The file unprov.tmpl was parsed successfully in 375 ms.
The parser initialization time was 63 ms.
The parser parse time was 312 ms.
```

コマンドラインでのマクロ変数の指定

runCfgUtil.sh コマンドを使用して、マクロ変数を指定します。

シンタックスの説明　***runCfgUtil.sh -e -l*** *file* ***-m*** *"macros"*

- ***-e***：符号化オプションを示します。
- *-l*：入力ファイルがローカル ファイル システムにあることを指定します。
- *file*：解析する入力テンプレート ファイルを示します。
- ***-m***：テンプレートを解析するときに代入するマクロ変数を指定します。
- *"macros"*：目的のマクロを示します。複数のマクロ変数が必要な場合は、各マクロの間に 2 つのカンマを挿入します。

コマンドラインでマクロ変数の値を指定するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/rdu/templates* にディレクトリを変更します。

ステップ 2　使用するテンプレート ファイルを選択します。

ステップ 3　テンプレートのマクロ変数を特定します。この例のマクロ変数は、macro1（option 3）と macro11（option 4.2）です。

ステップ 4　マクロ変数の値を特定します。macro1 の値を 1 に設定し、macro11 の値を 64000 に設定します。

ステップ 5　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -e -l macro.tmpl -m "macro1=1,,macro11=64000"
```

- **macro.tmpl**：入力ファイルを示します。
- **macro1=1,,macro11=64000**：マクロ変数のキーと値のペアを示します。複数のマクロ変数が必要なため、キーと値のペアの間に 2 つのカンマを挿入して区切ります。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes       Option   Description                      Value
0       030101           3        Network Access Control           On
3       041F             4        Class of Service
5       010101           4.1      Class ID                         1
8       02040000FA00     4.2      Maximum Downstream Rate          64000 bits/sec
14      03040000FA00     4.3      Maximum Upstream Rate            64000 bits/sec
20      040101           4.4      Upstream Channel Priority        1
...

0 error(s), 0 warning(s) detected. Parsing of macro.tmpl was successful.
The file macro.tmpl was parsed successfully in 854 ms.
The parser initialization time was 76 ms.
The parser parse time was 778 ms.
```

マクロ変数のデバイスの指定

runCfgUtil.sh コマンドを使用して、マクロ変数のデバイスを指定します。

シンタックスの説明　***runCfgUtil.sh -e -r*** *file* ***-i*** *MAC* ***-u*** *username* ***-p*** *password*

- ***-e***：符号化オプションを示します。
- *-r*：入力ファイルが RDU に追加したファイルであることを示します。
- *file*：解析する入力テンプレート ファイルを示します。
- ***-i***：マクロ変数を解析するときに使用するデバイスを指定します。
- *MAC*：デバイスの MAC アドレスを示します。
- ***-u*** *username*：RDU に接続するときに使用するユーザ名を指定します。
- ***-p*** *password*：RDU に接続するときに使用するパスワードを指定します。

マクロ変数の代入に使用するデバイスを指定するには、次の手順に従います。

ステップ 1　使用するテンプレート ファイルを選択します。この例では、既存のテンプレート ファイル *macro.tmpl* を使用します。

ステップ 2　テンプレートのマクロ変数を特定します。この例のマクロ変数は、macro1（option 3）と macro11（option 4.2）です。

ステップ 3　使用するデバイスを調べます。この例では、デバイスが RDU に存在し、マクロ変数がプロパティとして設定されているものとします。macro1 の値を 1 に設定し、macro11 の値を 64000 に設定します。

ステップ 4　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -e -r macro.tmpl -i "1,6,00:01:02:03:04:05" -u admin
-p changeme
```

- **macro.tmpl**：入力ファイルを示します。
- **1,6,00:01:02:03:04:05**：デバイスの MAC アドレスを示します。この MAC アドレスは、例として示す目的でのみ使用しています。
- **admin**：デフォルトのユーザ名を示します。
- **changeme**：デフォルトのパスワードを示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes        Option     Description                     Value
0       030101            3         Network Access Control           On
3       041F              4         Class of Service
5       010101            4.1       Class ID                         1
8       02040000FA00      4.2       Maximum Downstream Rate          64000 bits/sec
14      03040000FA00      4.3       Maximum Upstream Rate            64000 bits/sec
20      040101            4.4       Upstream Channel Priority        1
...

0 error(s), 0 warning(s) detected. Parsing of macro.tmpl was successful.
The file macro.tmpl was parsed successfully in 159 ms.
The parser initialization time was 42 ms.
The parser parse time was 117 ms.
```

バイナリ ファイルへの出力の指定

runCfgUtil.sh コマンドを使用して、解析するテンプレートの出力をバイナリ ファイルとして指定します。

シンタックスの説明　***runCfgUtil.sh -l*** *input_file* ***-o*** *output_file*

- ***-l***：入力ファイルがローカル ファイル システムにあることを指定します。
- *input_file*：解析する入力テンプレート ファイルを示します。
- ***-o***：解析するテンプレート ファイルをバイナリ ファイルとして保存することを指定します。
- *output_file*：解析するテンプレート ファイルのバイナリ コンテンツを格納するファイル名を示します。

テンプレートを解析してバイナリ ファイルに出力するように指定するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/rdu/templates* にディレクトリを変更します。

ステップ 2　使用するテンプレート ファイルを選択します。

ステップ 3　出力ファイルの名前を指定します。この例では *unprov.cm* を使用します。

ステップ 4　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -l unprov.tmpl -o unprov.cm
```

- **unprov.tmpl**：バイナリ ファイルに解析結果を出力する、既存のテンプレート ファイルを示します。
- **unprov.cm**：使用する出力ファイル名を示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0
0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.
The file unprov.tmpl was parsed successfully in 595 ms.
The parser initialization time was 262 ms.
The parser parse time was 333 ms.
```

ローカル バイナリ ファイルの表示

runCfgUtil.sh コマンドを使用して、ローカル システムに格納されているバイナリ ファイルを表示します。

シンタックスの説明　***runCfgUtil.sh -d -l*** *file*

- ***-d***：このコマンドでバイナリ入力ファイルを表示するためにデコードすることを指定します。
- ***-l***：入力ファイルがローカル ファイル システムにあることを示します。
- *file*：表示する既存のバイナリ入力ファイルを示します。

ローカル ファイル システムにあるバイナリ ファイルを表示するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/rdu/samples/packet_cable* にディレクトリを変更します。

ステップ 2　表示するバイナリ ファイルを選択します。

ステップ 3　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -d -l unprov_packet_cable.bin
```

unprov_packet_cable.bin：表示する既存のバイナリ入力ファイルを示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26

Warning: Expecting config file of type docsis, but input file is of type pktc1.0.
Decoding as pktc1.0
Off     File Bytes        Option   Description            Value
0       FE0101            254      Telephony Config File  1
                                   Start/End

        0B153013060E      11       SNMP MIB Object        .iso.org.dod.internet.privat
3       2B06010401A30B02                                   e.enterprises.cableLabs.clab
        0201010107000201                                   Project.clabProjPacketCable.
        02                                                 pktcMtaMib.pktcMtaMibObjects
                                                           .pktcMtaDevBase.pktcMtaDevEn
                                                           abled.0,INTEGER,fals e(2)
...
```

（注） この例の警告は、デフォルトの入力ファイルが DOCSIS で、この例ではバイナリ PacketCable ファイルを使用しているために表示されます。**-pkt** オプションを使用して、入力ファイルを PacketCable ファイルとして指定する場合、警告は表示されません。次に例を示します。

```
/opt/CSCObac/rdu/bin/# runCfgUtil.sh -d -pkt -l unprov_packet_cable.bin
```

外部バイナリ ファイルの表示

runCfgUtil.sh コマンドを使用して、外部バイナリ ファイルを表示します。

シンタックスの説明　***runCfgUtil.sh -d -r*** *file* ***-u*** *username* ***-p*** *password*

- ***-d***：このコマンドでバイナリ入力ファイルを表示するためにデコードすることを指定します。
- ***-r***：入力ファイルが RDU に追加したファイルであることを示します。
- *file*：RDU にある既存のバイナリ ファイルを示します。
- ***-u*** *username*：RDU に接続するときに使用するユーザ名を指定します。
- ***-p*** *password*：RDU に接続するときに使用するパスワードを指定します。

RDU に追加したバイナリ ファイルを表示するには、次の手順に従います。

ステップ 1　表示するバイナリ ファイルを選択します。この例では、既存のバイナリ ファイル *unprov.cm* を使用し、RDU が localhost:49187 であることを前提にしています。

ステップ 2　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -d -r unprov.cm -u admin -p changeme
```

- **unprov.cm**：RDU にある既存のバイナリ ファイルを示します。
- **admin**：デフォルトのユーザ名を示します。
- **changeme**：デフォルトのパスワードを示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes       Option     Description                      Value
0       030100           3         Network Access Control            Off
3       041F             4         Class of Service
5       010101           4.1       Class ID                          1
8       02040001F400     4.2       Maximum Downstream Rate           128000 bits/sec
14      03040000FA00     4.3       Maximum Upstream Rate             64000 bits/sec
20      040101           4.4       Upstream Channel Priority         1
...
252     06108506547F     6         CM MIC Configuration Setting      8506547FC9152B44
        C9152B44DB95                                                 DB955420843EF6FE
        5420843EF6FE
270     0710644B675B     7         CMTS MIC Configuration Setting    644B675B70B7BD3E
        70B7BD3E09AC                                                 09AC210F794A1E8F
        210F794A1E8F
288     FF               255       End-of-Data Marker
289     00               0         PAD
290     00               0         PAD
291     00               0         PAD
0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.
The file unprov.tmpl was parsed successfully in 375 ms.
The parser initialization time was 63 ms.
The parser parse time was 312 ms.
```

PacketCable Basic フローの有効化

runCfgUtil.sh コマンドを使用して、PacketCable Basic フローの完全性ハッシュを生成し、BASIC フローの静的設定ファイルに挿入することをサポートします。

シンタックスの説明　***runCfgUtil.sh -t*** {***basic*** | ***secure***} ***-pkt -r*** *filename* ***-u*** *username* ***-p*** *password*

- ***basic***：PacketCable Basic フローの完全性ハッシュを計算し、MTA 静的設定ファイルに挿入します。
- ***secure***：PacketCable Basic フローの完全性ハッシュを MTA 静的設定ファイルに挿入しません。これはデフォルト設定です。
- ***-r***：入力ファイルが RDU に追加したファイルであることを示します。
- *filename*：入力ファイルを示します。
- ***-u*** *username*：RDU に接続するときに使用するユーザ名を指定します。
- ***-p*** *password*：RDU に接続するときに使用するパスワードを指定します。
- ***-pkt***：入力ファイルが PacketCable MTA 設定ファイルであることを示します。

PacketCable Basic フローの完全性ハッシュを生成し、Basic フローの静的設定ファイルに挿入することをサポートするには、次の手順に従います。

ステップ 1　PacketCable Basic フローの完全性ハッシュを挿入する Basic フローの静的設定ファイルを選択します。この例では *example_mta_config.tmpl* を使用します。

ステップ 2　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -t basic -pkt -r example_mta_config.tmpl -u admin
-p changeme
```

- **example_mta_config.tmpl**：Basic フローの静的設定ファイルを示します。
- **admin**：デフォルトのユーザ名を示します。
- **changeme**：デフォルトのパスワードを示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off     File Bytes      Option   Description              Value
0       FE0101          254      Telephony Config File    1
                                 Start/End
3       0B153013060E    11       SNMP MIB Object          .iso.org.dod.internet.privat
        2B06010401A3                                      e.enterprises.cableLabs.clab
        0B0202010101                                      Project.clabProjPacketCable.
        0700020101                                        pktcMtaMib.pktcMtaMibObjects
                                                          .pktcMtaDevBase.pktcMtaDevEn
                                                          abled.0,INTEGER,true(1)
26      0B2530230610    11       SNMP MIB Object          .iso.org.dod.internet.privat
        2B06010401A3                                      e.enterprises.cableLabs.clab
        0B0202020102                                      Project.clabProjPacketCable.
        01010109040F                                      pktcSigMib.pktcSigMibObjects
        434D532E4950                                      .pktcNcsEndPntConfigObjects.
        464F4E49582E                                      pktcNcsEndPntConfigTable.pkt
        434F4D                                            cNcsEndPntConfigEntry.pktcNc
                                                          sEndPntConfigCallAgentId.9,S
                                                          TRING,CMS.IPFONIX.COM
...
371     FE01FF          254      Telephony Config File    255
                                 Start/End
0 error(s), 0 warning(s) detected. Parsing of example_mta_config.tmpl was successful.
The file example_mta_config.tmpl was parsed successfully in 100 ms.
The parser initialization time was 44 ms.
The parser parse time was 56 ms.
```

.tmpl 拡張子付きのファイルは、動的設定テンプレートであると見なされます。このテンプレートに対し、テンプレート処理中に Basic ハッシュの計算と挿入が透過的に発生します。その結果、同じテンプレートを Secure モードと Basic モードのプロビジョニングで使用できます。

ただし、ハッシュを挿入する前に、Secure 静的バイナリ設定ファイルを Basic 静的設定ファイルに変換する場合は、次の手順に従います。

a.　次のコマンドを使用して、Secure 静的ファイルをテンプレートに変換します。

runCfgUtil -l *input_static_filename* ***-pkt -g -o*** *output_template_filename*

b.　次のコマンドを使用して、Secure 静的テンプレートを Basic 静的設定ファイルに変換します。

runCfgUtil -t basic -l *input_template_name* ***-pkt -o*** *output_Basic_static_filename*

このコマンドは、Basic の完全性ハッシュを計算して、Basic 静的設定ファイルに挿入します。

マルチベンダーをサポートするための TLV 43 の生成

runCfgUtil.sh コマンドを使用して、マルチベンダーのサポートを提供するために TLV 43 を生成します。

シンタックスの説明　***runCfgUtil.sh -docsis -r*** *filename* ***-u*** *username* ***-p*** *password*

- ***-docsis***：入力ファイルが DOCSIS テンプレート ファイルであることを示します。
- *filename*：解析する入力テンプレート ファイルを示します。
- **-r**：入力ファイルが RDU に追加したファイルであることを示します。
- ***-u*** *username*：RDU に接続するときに使用するユーザ名を指定します。
- ***-p*** *password*：RDU に接続するときに使用するパスワードを指定します。

RDU に追加したテンプレート ファイルを使用して TLV 43 を生成するには、次の手順に従います。

ステップ 1　使用するテンプレート ファイルを選択します。この例では、既存のテンプレート ファイル *test.tmpl* を使用します。DOCSIS テンプレートを使用するため、**-docsis** オプションを使用します。

ステップ 2　次のコマンドを使用して、設定ファイル ユーティリティを実行します。

```
/opt/CSCObac/rdu/bin# runCfgUtil.sh -docsis -r test.tmpl -u admin -p changeme
```

- ***test.tmpl***：DOCSIS 設定ファイルを示します。
- **admin**：デフォルトのユーザ名を示します。
- **changeme**：デフォルトのパスワードを示します。

ユーティリティを実行すると、次のような結果が表示されます。

```
Broadband Access Center Configuration Utility
Version: 4.0, Revision: 1.26
Off   File Bytes      Option    Description                       Value
0     2B14            43        DOCSIS Extension Field
2     0803FFFFFF      43.8      Vendor ID                         FF-FF-FF
7     050D            43.5      L2VPN Encoding
9     010502345600    43.5.1    VPNID Subtype                     0234560003
      03
16    0204            43.5.2    NSI Encapsulation Subtype
18    02020019        43.5.2.2  IEEE 802.1Q Format Subtype        25
22    2B0B            43        DOCSIS Extension Field
24    080300000C      43.8      Vendor ID                         00-00-0C (CISCO
                                                                  SYSTEMS, INC.)
29    010418017A30    43.1      Static Downstream Frequency       402750000
35    2B1D            43        DOCSIS Extension Field
37    080300000C      43.8      Vendor ID                         00-00-0C (CISCO
                                                                  SYSTEMS, INC.)
42    0316626F6F74    43.3      Update Boot Monitor Image         boot_monitor_image.bin
      5F6D6F6E6974
      6F725F696D61
      67652E62696E
66    061071E79068    6         CM MIC Configuration Setting      71E790683DE8B995
      3DE8B9950536                                                05368936F4C5312F
      8936F4C5312F
84    0710DB0EED14    7         CMTS MIC Configuration Setting    DB0EED14B5B3428D
      B5B3428D2B15                                                2B150DA582B41A54
      0DA582B41A54
102   FF              255       End-of-Data Marker
103   00              0         PAD
0 error(s), 0 warning(s) detected. Parsing of test.tmpl was successful.
The file test.tmpl was parsed successfully in 250 ms.
The parser initialization time was 109 ms.
The parser parse time was 141 ms.
```

CHAPTER 6

DOCSIS 設定

この章では、Broadband Access Center（BAC）DOCSIS 配備のプロビジョニング フローについて説明します。また、設定に先立って必要な情報を説明し、使用可能なツールについて説明します。

（注）この BAC リリースでサポートされている DOCSIS オプションについては、P.B-2 の「DOCSIS オプションのサポート」を参照してください。

この章は、ユーザが次の仕様の内容を熟知していることを想定しています。

- DOCSIS 3.0：
  - CM-SP-SECv3.0-I04-070518
  - CM-SP-PHY3.0-I04-070518
  - CM-SP-MULPIv3.0-I04-070518
  - CM-SP-OSSIv3.0-I03-070518
- DOCSIS 2.0：
  - CM-SP-RFI2.0-I12-071206
  - L2VPN CM-SP-L2VPN-I06-071206

DOCSIS ワークフロー

この項では、DHCPv4 と DHCPv6 の DOCSIS プロビジョニング仕様に含まれるプロビジョニング ワークフローについて説明します。

- DOCSIS DHCPv4 ワークフロー（P.6-2）
- DOCSIS DHCPv6 ワークフロー（P.6-5）

DOCSIS DHCPv4 ワークフロー

図 6-1 に、DHCPv4 の DOCSIS プロビジョニング仕様に含まれるプロビジョニング ワークフローを示します。その後、各ステップについて説明します。

図 6-1　DOCSIS DHCPv4 プロビジョニング フロー

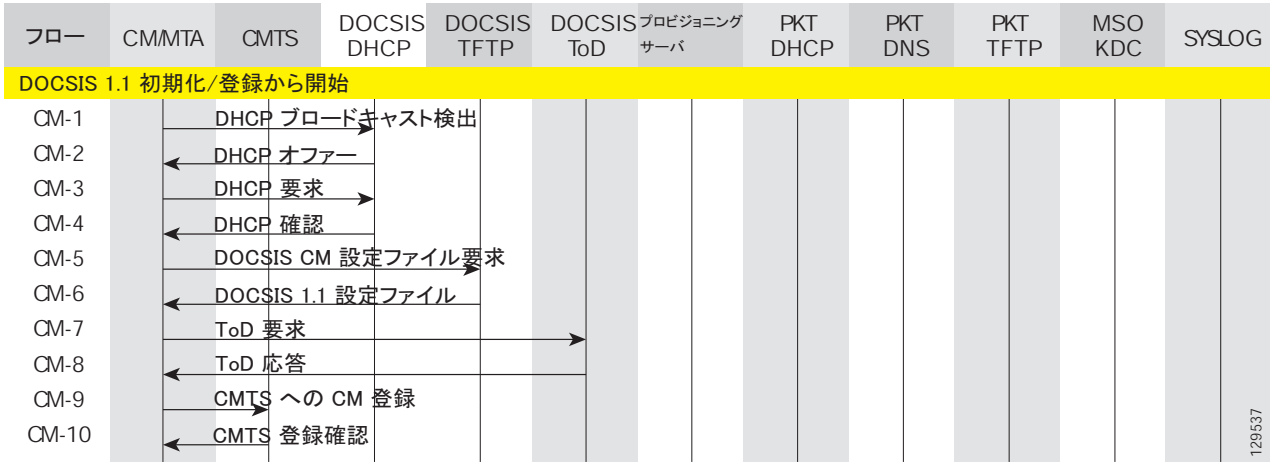


表 6-1 で、図 6-1 に示されるさまざまな DOCSIS プロビジョニング ステップで発生する可能性がある問題について説明します。

表 6-1　DOCSIS DHCPv4 ワークフローの説明

手順	DOCSIS DHCPv4 ワークフロー	潜在的な問題
CM[1] -1	DHCP 検出	• init(d) 状態。 • 使用可能なアドレスがない。 • BAC 共有秘密情報に誤りがある。 • サービス クラスが誤って設定される。 • DOCSIS テンプレートの解析エラー（無効なオプション、インクルード ファイルが見つからないなど）。 **Cisco Network Registrar DHCP** • DHCP 設定に誤りがある。 • DHCP サーバがプロビジョニング グループ内に存在しない。 **BAC Network Registrar 拡張** • Network Registrar 拡張が DPE に接続できない。 • Network Registrar 拡張がプロビジョニング グループ内の DPE を検出できない。 • 拡張が RDU に接続されていることを確認する。 • Network Registrar 拡張が DPE キャッシュ ミスを取得し、要求を RDU に送信する。 **RDU** • 適切なスコープが定義されていない（または RDU の設定と一致しない）。 • RDU の IP アドレスに誤りがある。 • RDU ポートに誤りがある（デフォルト 49187）。 • DPE から RDU に PING できない。 • RDU で構成の生成に失敗している。 • RDU ライセンス数が超過したため、設定されない。 • RDU でデバイスの検出に失敗している。 **DPE** • DPE がプロビジョニング グループに割り当てられていない。 • DHCP サーバから DPE に PING できない。 • DPE　インターフェイスでプロビジョニングがイネーブルになっていない。
CM-2	DHCP オファー	DHCP と Cable Modem Termination System（CMTS; ケーブル モデム ターミネーション システム）間のルーティングの問題。
CM-3	DHCP 要求	• init(i) 状態。 • DHCP サーバからすべての必須パラメータが提供されなかった。
CM-4	DHCP 確認	

表 6-1　DOCSIS DHCPv4 ワークフローの説明（続き）

手順	DOCSIS DHCPv4 ワークフロー	潜在的な問題
CM-5	TFTP 要求	• Init(o) 状態。 • CMTS と DPE 間のルーティングの問題。 • TFTP サーバ（DPE）からモデムへのルートがない。 • DPE キャッシュ ミス（静的ファイル、RDU がダウンしているかファイルがない）。 • TFTP サーバ（DPE）でファイルが見つからない。 • DPE キャッシュ ミス（動的ファイル）。 • DPE IP 検証エラー（たとえば、デバイスの IP アドレスが予期したアドレスではない、動的共有秘密が CMTS でイネーブルになっている、DOCSIS モデムとしてハッカーがスプーフィングしている）。
CM-6	TFTP 応答	DPE と CMTS 間のルーティングの問題。
CM-7	ToD 要求	init(t) 状態：タイム サーバ（DPE）からモデムへのルートがない。
CM-8	ToD 応答	
CM-9	CMTS への CM の登録	• reject(m)：* CMTS 共有秘密情報が BAC または DPE DOCSIS 共有秘密情報と一致しない。 • reject(c)：* 誤った DOCSIS 設定ファイルが配信された（1.1 ファイルを 1.0 ケーブル モデムへ配信）。
CM-10	CMTS 登録確認応答	許容できる状態： • online • online(d) • online(pk) • online(pt)

1. CM = ケーブル モデム

DOCSIS DHCPv6 ワークフロー

図 6-2 に、DHCPv6 の DOCSIS プロビジョニング仕様に含まれるプロビジョニング ワークフローを示します。その後、各ステップについて説明します。

図 6-2　DOCSIS DHCPv6 プロビジョニング フロー

DHCPv6 の DOCSIS プロビジョニング ワークフローには、次の割り当てを含む、ケーブル モデムによる IPv6 接続の確立が含まれます。

- リンクローカル アドレス
- デフォルト ルータ
- IPv6 管理アドレス
- その他の IPv6 の設定

表 6-2 で、図 6-2 に示されるさまざまな DOCSIS プロビジョニング ステップで発生する可能性がある問題について説明します。

表 6-2　DOCSIS DHCPv6 ワークフローの説明

ワークフロー	説明	潜在的な問題
プロビジョニング段階：リンクローカル アドレスの割り当て		
ケーブル モデムは、インターフェイスの MAC アドレスから取得する EUI-64（64 ビット Extended Unique Identifier）から IPv6 リンクローカル アドレスを構築します。		
NS（DAD）	ケーブル モデムは、ネイバー送信要求（NS）メッセージを使用して、Duplicate Address Detection（DAD; 重複アドレス検出）を実行します。DAD は、構築したリンクローカル アドレスがすでに使用中であるかどうかを確認します。NS に対する応答がない場合、ケーブル モデムはリンクローカル アドレスが未使用であると判断します。応答が返ってきた場合は、リンクローカル アドレスと MAC アドレスが競合していることを示し、ケーブル モデムはプロビジョニング プロセスを停止します。	
プロビジョニング段階：ルータ検出		
ケーブル モデムは、ルータ検出を使用してデフォルトのルータを検出し、HFC リンクのプレフィックスを特定します。		
RS	ケーブル モデムは、ルータ送信要求（RS）を CMTS に送信して、定期的に ルータ アドバタイズメント（RA）メッセージの転送をトリガーします。	
RA	CMTS ルータは、定期的に RA を送信します。各 RA には次のものが含まれます。 • リンクに割り当てる IPv6 プレフィックスのリスト • DHCPv6 を使用するためのディレクティブ • デフォルト ルータとしての CMTS ルータのアベイラビリティ	

表 6-2　DOCSIS DHCPv6 ワークフローの説明（続き）

ワークフロー	説明	潜在的な問題
プロビジョニング段階：DHCPv6		
送信要求	ケーブル モデムは、送信要求メッセージを送信して DHCP サーバを検索します。	• init6(s) 状態。 • 使用可能な IPv6 アドレスがない。 • BAC 共有秘密情報に誤りがある。 • サービス クラスが誤って設定される。 • DOCSIS テンプレートの解析エラー（無効なオプション、インクルード ファイルが見つからないなど）。 **Network Registrar DHCP** • DHCPv6 設定に誤りがある。 • DHCP サーバがプロビジョニング グループ内に存在しない。 • 適切なプレフィックスが定義されていない（または BAC RDU 設定と一致しない）。 **BAC Network Registrar 拡張** • Network Registrar 拡張が DPE に接続できない。 • Network Registrar 拡張がプロビジョニング グループ内の IPv6 DPE を検出できない。 • 拡張が RDU に接続されていることを確認する。 • Network Registrar 拡張が DPE キャッシュ ミスを取得し、要求を RDU に送信する。 **RDU** • RDU の IP アドレスに誤りがある。 • RDU ポートに誤りがある（デフォルト 49187）。 • DPE から RDU に PING できない。 • RDU で構成の生成に失敗している。 • RDU ライセンス数が超過したため、設定されない。 • RDU でデバイスの検出に失敗している。 **DPE** • DPE がプロビジョニング グループに割り当てられていない。 • DHCP サーバから DPE に PING できない。 • DPE インターフェイスで IPv6 プロビジョニングがイネーブルになっていない。 • IPv6 プロビジョニングのプロビジョニング グループがイネーブルになっていない。

表 6-2　DOCSIS DHCPv6 ワークフローの説明（続き）

ワークフロー	説明	潜在的な問題
Relay-Forw	リレー エージェントは、ケーブル モデムから受信する DHCPv6 メッセージ全体を DHCPv6 サーバに転送します。 リレー エージェントは、次のようなリレー エージェントのメッセージ フィールドとオプションを追加します。 • Peer-address • Link-address • Interface ID	
Relay-Repl	リレー エージェントは、サーバの応答を抽出して、CMTS 経由でケーブル モデムに転送します。	
アドバタイズ	DHCP サーバは、ケーブル モデムから受信した送信要求メッセージに対する応答として、DHCP サービスでサーバを利用可能なことを示すアドバタイズ メッセージを返します。	• init6(a) 状態。 • DHCP と CMTS 間のルーティングの問題。
要求	アドバタイズ メッセージを受信すると、ケーブル モデムは要求メッセージを送信して、特定のサーバの IP アドレスを含む、構成パラメータを要求します。	• init6(r) 状態。 • DHCP サーバからすべての必須パラメータが提供されなかった。
Relay-Forw	リレー エージェントは、メッセージを DHCPv6 サーバに転送します。	
Relay-Repl	リレー エージェントは、サーバの応答を抽出して、CMTS 経由でケーブル モデムに転送します。	
応答	CMTS は、割り当て済みアドレスと構成パラメータを含む、DHCP サーバから受信した応答メッセージを転送します。	init6(i) 状態。

（注） DHCPv6 クライアントは、Rapid Commit モードでプロビジョニングできます。Rapid Commit は、通常の 4 つのメッセージ交換の代わりに、2 つのメッセージ交換を行います。2 つのメッセージ交換には、送信要求と応答が含まれます。4 つのメッセージ交換には、送信要求、アドバタイズ、要求、応答が含まれます。これらメッセージはすべて、リレー エージェントを通過する場合、Relay-Forw または Relay-Repl メッセージにラップされます。

Rapid Commit がイネーブルになっている場合は、DHCP サーバは送信要求（Relay-Forw メッセージにラップ）メッセージに対し 応答（Relay-Repl メッセージにラップ）メッセージで応答します。Rapid Commit をディセーブルにすると、DHCP サーバは アドバタイズ（Relay-Reply にラップ）メッセージで応答します。

NS（DAD）	DHCPv6 メッセージの交換が完了すると、ケーブル モデムは、リンクローカル アドレスが DAD 経由ですでに使用中であるかどうかを確認します。応答を受信しない場合、ケーブル モデムは IP アドレスの取得に成功したと判断します。	
プロビジョニング段階：ToD		
要求	IPv6 アドレスを取得後、ケーブル モデムは RFC 868 タイム サーバの Time of Day を要求します。	init6(t) 状態：タイム サーバ（DPE）からモデムへのルートがない。
応答	サーバの IPv6 アドレスは、DHCPv6 オプションを使用して指定します。	

表 6-2　DOCSIS DHCPv6 ワークフローの説明（続き）

ワークフロー	説明	潜在的な問題
プロビジョニング段階：TFTP		
TFTP 取得	ケーブル モデムは、TFTP を使用して設定ファイルをダウンロードします。サーバの IPv6 アドレスと設定ファイルの名前は、DHCPv6 経由で利用可能になります。	• init6(o) 状態。 • CMTS と DPE 間のルーティングの問題。 • TFTP サーバ（DPE）からモデムへのルートがない。 • DPE キャッシュ ミス（静的ファイル、RDU がダウンしているかファイルがない）。 • TFTP サーバ（DPE）でファイルが見つからない。 • DPE キャッシュ ミス（動的ファイル）。 • DPE IP 検証エラー（たとえば、デバイスの IP アドレスが予期したアドレスではない、Dynamic Shared Secret が CMTS でイネーブルになっている、DOCSIS モデムとしてハッカーがスプーフィングしている）。
TFTP RSP（設定ファイル）		DPE と CMTS 間のルーティングの問題。
これで、ケーブル モデムが IPv6 操作用にプロビジョニングされました。		

動的 DOCSIS テンプレートによる MIB の使用方法

BAC に付属する MIB の完全なリストについては、P.5-7 の「SNMP VarBind」を参照してください。

RDU にロードされる DOCSIS MIB には、2 つのバージョンがあります。

- DOCS-CABLE-DEVICE-MIB-OBSOLETE（実験的ブランチ）
- DOCS-CABLE-DEVICE-MIB（mib2 ブランチ）

これらの使用方法については、P.5-7 の「DOCSIS MIB」を参照してください。

Application Programming Interface（API; アプリケーション プログラミング インターフェイス）コールを使用するか、*rdu.properties* を修正することにより、MIB を追加することができます。詳細については、P.7-34 の「Euro-PacketCable MIB の設定」を参照してください。

次の場合に、SNMP TLV をテンプレートに追加できます。

- 利用可能な MIB がないとき。P.5-11 の「MIB を使用しない SNMP TLV の追加」を参照してください。
- ベンダー固有の MIB を使用するとき。P.5-12 の「ベンダー固有の MIB を使用した SNMP TLV の追加」を参照してください。

DOCSIS 設定のための BAC 機能

この項では、DOCSIS テクノロジーと関連する BAC の付加価値機能について説明します。

動的設定 TLV

DPE は、動的 DOCSIS 設定の TFTP 要求を受け取るたびに、次の TLV を追加します。

- TLV 19：TFTP サーバのタイムスタンプ（オプション）：Configure DOCSIS Defaults ページに TFTP Time Stamp Option として表示されます。詳細については、表 13-3 を参照してください。この TLV では、CMTS および DPE における NTP の同期が必要です。
- TLV 20 および TLV 59：IPv4 と IPv6 用の TFTP サーバのプロビジョニングされたモデム アドレス（オプション）：Configure DOCSIS Defaults ページに TFTP Modem Address Option として表示されます。詳細については、表 13-3 を参照してください。

（注） DPE の TFTP IP 検証機能は、Cisco CMTS DSS 機能とは互換性がありません。P.6-10 の「DPE TFTP IP 検証」を参照してください。Cisco CMTS に DSS が設定されている場合、TFTP サーバのプロビジョニングされたモデム アドレスをディセーブルにする必要があります。

- TLV 6：CM MIC の設定（必須）
- TLV 7：CMTS MIC の設定（必須）：Configure DOCSIS Defaults ページに CMTS Shared Secret として表示されます。詳細については、表 13-3 を参照してください。
- TLV 43.6.x：拡張 CMTS MIC の設定（必須）：Configure DOCSIS Defaults ページに CMTS Shared Secret として表示されます。詳細については、表 13-3 を参照してください。

（注） CMTS MIC を設定するときは、次の CMTS IOS リリースの依存関係に注意してください。

- TLV 39 または TLV 40 を含める場合、DOCSIS 2.0 CMTS MIC には CMTS IOS 12.3BC が必要です。
- 次の CMTS IOS コマンドが BAC によって設定されているものとします。
  - **ip dhcp relay information option**
  - **no ip dhcp relay information check**
  - **cable helper-address** *x.x.x.x*

    ここで、*x.x.x.x* は Network Registrar DHCP サーバの IP アドレスです。

    IPv6 環境では、**cable helper-address** の代わりに次のコマンドを使用する必要があります。**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*]
  - **cable dhcp-giaddr primary**

DPE TFTP IP 検証

DPE TFTP サーバは、動的設定ファイルを対象に、TFTP クライアントの IP アドレスが DOCSIS ケーブル モデムの予期した IP アドレスと一致することを確認します。一致しない場合、要求は破棄されます。この機能に Cisco CMTS DMIC 機能との互換性はありません。

TFTP の動的構成要求における要求者の IP アドレスの検証をディセーブルにするには、**no service tftp** *1..1* **ipv4 | ipv6 verify-ip** コマンドを使用します。詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

DOCSIS 1.0、1.1、2.0、および 3.0 のサポート

BAC 4.0 は、DOCSIS 1.0、1.1、2.0、および 3.0 をサポートします。TLV の詳細については、P.5-2 の「テンプレート文法」を参照してください。この BAC リリースがサポートする各 DOCSIS バージョンのオプションのリストについては、P.B-2 の「DOCSIS オプションのサポート」を参照してください。

DOCSIS バージョンの動的選択

BAC は、ケーブル モデムの DOCSIS バージョンを着信 DHCP 要求から検出できます。また、次の 2 つの方法のいずれかで CMTS の DOCSIS バージョンを検出することもできます。

- DHCPv4 の Option 82 と DHCPv6 の Option 17 を使用して、CMTS をその DOCSIS バージョンを送信するリレー エージェントとして使用する方法。
- GIADDR から CMTS DOCSIS バージョンへのマッピングを行う顧客提供のソースを使用する方法。

この DOCSIS バージョンを使用して、BAC は、デバイスの最適な DOCSIS 設定ファイルを判断します（そのように設定されている場合）。このバージョンが、デバイスと CMTS 間で最も一般的な DOCSIS バージョンです。たとえば、デバイスが DOCSIS 2.0 をサポートし、CMTS が DOCSIS 1.1 をサポートする場合、DOCSIS 1.1 ファイルが使用されます。

モデムの DOCSIS バージョンの判別

BAC は、ケーブル モデムの DOCSIS バージョンを着信 DHCP 要求から検出できます。この要求のモデムの機能を示す Vendor Class Identifier フィールド（Option 60）に文字列が含まれています。たとえば、「docsis1.1:*xxxxxx*」では、*xxxxxx* はモデムの機能を表す ASCII 文字列です。サービス レベル選択拡張は、「docsis」と「*:xxxxxxx*」16 進文字列間の文字をモデムの DOCSIS バージョンとして使用します。

CMTS の DOCSIS バージョンの判別

この BAC リリースでは、CMTS をイネーブルにしてリレー エージェントとして動作させ、CMTS の DOCSIS バージョンを提供できます。この機能は次のオプションを使用してイネーブルにします。

- DHCPv4 Relay Agent Option 82。このオプションを使用すると、CMTS は CMTS の特定機能を送信（またはアドバタイズ）できます。このオプションは、DOCSIS DHCP ベンダー指定のオプションで、CMTS の DOCSIS バージョンを伝達します。
- DHCPv6 Vendor-specific Information Option 17。このオプションを使用すると、ベンダー固有の情報を指定できます。このオプションは、Relay-forward メッセージと Relay-reply メッセージで伝達され、DHCPv6 リレー エージェントと DHCPv6 サーバ間で情報を送信します。

以前のバージョンと同様、この BAC バージョンは、DHCP GIADDR フィールドを介して CMTS の DOCSIS バージョンを判別します。このフィールドは、CMTS インターフェイスの IP アドレスを指定します。この方法の場合、DOCSIS モデムのサービス レベル選択拡張は、*/docsis/cmts/version/giaddrToVersionMap* プロパティを検索します。このプロパティの値は、GIADDR から DOCSIS バージョンへのマッピングを含んでいる外部ファイルの名前です。

このマッピング ファイルには *giaddr-docsis-map.txt* と名前を付けて、RDU に追加する必要があります。次の方法で、*giaddr-docsis-map.txt* ファイルを RDU に追加できます。

- *Configuration.addFile()* コールを介した API。
- 管理者のユーザ インターフェイス（**Configuration** > **Files** を選択してアクセス）。P.13-19 の「ファイルの追加」を参照してください。

giaddr-docsis-map.txt ファイルには、次の形式で必要な情報を含める必要があります。

```
IPv4_dotted_decimal_address_string,DOCSIS_version_string
```

- *IPv4_dotted_decimal_address_string*：CMTS インターフェイスの IP アドレスを指定します。
- *docsis_version_string*：ケーブル モデムがサポートする DOCSIS バージョンを示します。

サービス レベル拡張は、DHCP パケットに含まれる GIADDR アドレスを使用して、CMTS の DOCSIS バージョンを検索します。マッピング ファイルに GIADDR がない場合、拡張は */docsis/cmts/version/default* プロパティの値をケーブル モデムの DOCSIS バージョンの値に使用します。このプロパティのデフォルト値は **1.0** です。

giaddr-docsis-map.txt ファイルを動的に更新するには、*replaceExternalFile* API または管理者のユーザ インターフェイスを使用してファイルを編集して RDU 内で置き換えます。

（注） DOCSIS バージョンの選択のプロパティがサービス クラスで指定されていない場合、元のファイルが使用され、ネットワーク全体で体系的なアップグレードが可能になります。

DOCSIS バージョンに基づいたサービス レベルの選択

モデムの DOCSIS バージョンと CMTS を判別後、サービス レベル選択拡張はサポートされる最小限の DOCSIS バージョンを決定して、サービス レベルに */docsis/version* プロパティを設定します。このプロパティの値は、DOCSIS バージョン文字列（1.1 など）に設定されます。

（注） DOCSIS バージョンは、設定ファイル ユーティリティを使用して指定できます。詳細については、P.5-23 の「設定ファイル ユーティリティの使用方法」を参照してください。このファイル ユーティリティが実行する機能は、RDU DOCSIS Version Selector 機能が CMTS でサポートされる最新の DOCSIS バージョンを判別する RDU の検証とは異なります。

DOCSIS バージョンに基づいた DOCSIS 設定ファイル

BAC は、DOCSIS バージョンを使用して、モデムに送信する DOCSIS 設定ファイルの名前を判別します。

次のサービス クラス プロパティが、BAC の管理者のユーザ インターフェイスと API でサポートされます。

```
/cos/docsis/file/1.0
/cos/docsis/file/1.1
/cos/docsis/file/2.0
/cos/docsis/file/3.0/IPv4
/cos/docsis/file/3.0/IPv6
```

DOCSIS 設定ファイル名を特定の DOCSIS バージョンと関連付けるために、これらのプロパティをオプションで DOCSIS サービス クラスに追加できます。これらの各プロパティを設定すると、既存の DOCSIS 設定ファイル名プロパティで行われるように、RDU がサービス クラスとプロパティ値で指定されたファイルとの間にデータベース関係を確立します。

DOCSIS バージョン プロパティが存在する場合、BAC はそのプロパティ値から得られる DOCSIS バージョン文字列を、DOCSIS 設定ファイル名を提供するプロパティの名前に追加してプロパティ名を作成します。

```
/cos/docsis/file/docsis_version_string
```

サービス レベル拡張は、モデムのプロパティ階層内でこのプロパティ名を検索します。DOCSIS バージョン プロパティが見つかった場合は、プロパティ値を DOCSIS 設定ファイル名として使用します。DOCSIS バージョン プロパティが見つからない場合、BAC は、DOCSIS バージョン サフィックスのない DOCSIS 設定ファイル名を使用して、デバイス構成で指定するファイル名を提供します。

IPv6 のサポート

この BAC リリースでは、CableLabs DOCSIS 標準の最新リビジョン、DOCSIS 3.0 をサポートしています。DOCSIS 3.0 標準には、以前の DOCSIS 標準に基づき構築された主要な新機能が導入されています。次の機能があります。

- IPv6 デバイスのプロビジョニング。次のものがあります。
  - DOCSIS 準拠のケーブル モデムと CMTS
  - コンピュータ
  - 進化する OpenCable Application Platform に基づく RNG-200 STB
  - 混在 IP モードの PacketCable Multimedia Terminal Adapter（MTA; マルチメディア ターミナル アダプタ）などの、各種 eSAFE（embedded Service/Application Functional Entities）デバイスのバリアント

  BAC は、TFTP サービスと ToD サービスに対する IPv6 サポートなど、IPv6 デバイスをプロビジョニングするために必要なサービスを提供します。BAC は IPv6 デバイスの設定ファイルも処理します。
- 拡張されたアドレス指定機能

  IPv6 の主な利点は、その拡張されたアドレス指定機能です。IPv6 アドレスでは、アドレス空間が 32 ビットから 128 ビットに拡大されており、事実上、無制限のネットワークとシステムを提供できます。
- ケーブル モデムの IPv6 プロビジョニングと管理。このプロビジョニング フローには、次のものが含まれます。
  - サポートされる IP モード：BAC は、IPv4、IPv6、またはデュアル スタック モード（IPv4 および IPv6 プロビジョニング モードで構成）で DOCSIS ケーブル モデムをプロビジョニングします。さまざまなモードの詳細については、P.6-15 の「シングル スタックとデュアル スタック」を参照してください。

  （注） ケーブル モデムは、IP のプロビジョニング モードに関係なく、IPv4 と IPv6 のトラフィックを転送できます。

  - DHCPv6：DOCSIS プロビジョニング フローは、IPv6 の DHCP（DHCPv6 としても知られる）の使用を指定します。DOCSIS の DHCPv6 プロビジョニング フローの詳細については、P.6-5 の「DOCSIS DHCPv6 ワークフロー」を参照してください。

BAC における IPv6 デバイスのプロビジョニングをイネーブルにする前に、システムで IPv6 をイネーブルにする必要があります。コンピュータで IPv6 のサポートをイネーブルにするには、*root* としてログインして、次のコマンドを入力します。

```
# ifconfig intf inet6 plumb up
```

intf は、IPv6 をイネーブルにするインターフェイスを示します。

（注）物理イーサネット インターフェイスに加えて、ループバック インターフェイスでも plumb を実行してください。次に例を示します。

```
# ifconfig bge0 inet6 plumb up
# ifconfig lo0 inet6 plumb up
```

その後、次のコマンドも実行します。

```
# /usr/lib/inet/in.ndpd
# touch /etc/hostname6.intf
```

intf は、IPv6 をイネーブルにするインターフェイスを示します。

次の各項では、BAC の IPv6 に関連する概念について説明します。

- IPv6 のアドレス指定（P.6-14）
- シングル スタックとデュアル スタック（P.6-15）
- IPv6 の DHCP オプション（P.6-15）
- 属性とオプション（P.6-15）

IPv6 のアドレス指定

IPv6 アドレスの長さは 128 ビットで、コロン（:）で区切られた一連の 16 ビットの 16 進数フィールドとして表現されます。16 進数の A、B、C、D、E、および F には、大文字と小文字の区別はありません。次に例を示します。

```
2031:0000:130f:0000:0000:09c0:876a:130b
```

このアドレス指定は、次のように短縮できます。

- フィールドの先頭の 0 はオプションであり、09c0 は 9c0、0000 は 0 と記述できます。
- 0 のフィールドが連続する場合は（フィールドの数にかかわらず）、:: と表現できますが、アドレス内で 1 回しか使用できません。これは、複数回使用すると、アドレス パーサーが 0 の各ブロックのサイズを特定できないという理由からです。したがって、前述のアドレスは次のように記述できます。

  ```
  2031:0:130f::09c0:876a:130b
  ```

二重コロンの短縮形を使用すると、多くのアドレスを短縮できます。たとえば、ff01:0:0:0:0:0:1 は ff01::1 になります。

リンクローカル アドレスには、リンクに限定されたスコープがあり、プレフィックス fe80::/10 を使用します。ループバック アドレスのアドレスは、::1 です。マルチキャスト アドレスは、プレフィックス ff00::/8 で識別されます（IPv6 にはブロードキャスト アドレスはありません）。

IPv6 での IPv4 互換アドレスは、:: のプレフィックスを付けた 10 進数の 4 つの IPv4 アドレスです。たとえば、::c0a8:1e01 として解釈される IPv4 アドレスは、::192.168.30.1 と記述できます。

シングル スタックとデュアル スタック

RFC 4213 は、ホストとルータ内の IPv4 と IPv6 の両方のインターネット プロトコルを全面的にサポートするための技術としてデュアル スタックを定義します。IPv4 と IPv6 の両方をサポートするネットワーク スタックは、デュアル スタックと呼ばれ、デュアル スタックを実装するホストは、デュアル スタック ホストと呼ばれます。

BAC は、次の IP モードでケーブル モデムをプロビジョニングします。

- IPv4 のみ：このモードでは、ケーブル モデムは DHCPv4 サーバに、IPv4 アドレスと関連の操作パラメータを要求します。
- IPv6 のみ：このモードでは、ケーブル モデムは DHCPv6 サーバに、IPv6 アドレスと関連の操作パラメータを要求します。モデムは IPv6 アドレスを使用して、現在の Time-of-Day と設定ファイルを取得します。
- デュアル スタック：このモードでは、ケーブル モデムは IPv6 と IPv4 の両アドレス、およびパラメータを、DHCPv6 と DHCPv4 経由でほぼ同時に取得し、IPv6 アドレスの使用優先順位を上げて、Time-of-Day と設定ファイルを取得します。

（注） BAC は、ケーブル モデムがプロビジョニングされている直近の IP モードで検出されたデータのみ保存します。そのため、デバイスを DHCPv4 でブートし、その後 DHCPv6 でブートすると、DHCPv6 のデータのみ検出されて保存されます。

IPv4 および IPv6 モードでプロビジョニングしている間、ケーブル モデムは、所定の時間に IP アドレス タイプ（v4 または v6）の 1 つでのみ動作します。このような理由で、プロビジョニングの IPv4 および IPv6 モードは、シングル スタック モードと呼ばれます。

デュアル スタック モードでは、IPv4 と IPv6 のアドレスを同時に使用してケーブル モデムを管理できます。このモードでは、モデムは動作可能になった後に 2 番目の IP アドレスを取得します。この機能を使用して、DOCSIS ネットワークでの IPv4 から IPv6 への能率的な移行を行えます。

IPv6 の DHCP オプション

DOCSIS 3.0 標準は、DHCPv4 と DHCPv6 の複数の新規オプションを定義します。DHCPv6 オプションは、DHCPv4 オプションは使用しません。DHCPv6 のオプションは固有の別オプションです。BAC がサポートする DHCPv6 オプションのリストについては、『*User Guide for Cisco Network Registrar 7.0*』を参照してください。

属性とオプション

この項では、BAC 4.0 が Network Registrar と通信するときに使用する属性とオプションについて説明します。

BAC は、Network Registrar にインストールされている DHCP 拡張を使用して、そのデータベース内の設定に基づき DHCP メッセージを操作します。これらの拡張を使用して、BAC は DHCP 要求から情報を取得し、DHCP 応答に値を設定します。このようにして、BAC はプロビジョニングするデバイスにカスタマイズされた設定を提供します。

このインタラクションを容易にするために、Network Registrar は、辞書のセットを BAC 拡張に公開します。BAC 拡張は、これらの辞書を使用して Network Registrar と対話します。

辞書には、要求辞書と応答辞書が使用する属性辞書、環境辞書、および通知辞書の 3 種類があります。

- 環境辞書：DHCP サーバが拡張と通信するために使用する辞書に含まれる属性を表します。
- 要求辞書：要求パケットの DHCP オプションと属性を表します。
- 応答辞書：応答パケットの DHCP オプションと属性を表します。
- 通知辞書：BAC 拡張と RDU 間で伝達される情報を表します。

辞書は、BAC と Network Registrar で設定されているさまざまな 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ている DHCP 設定パラメータと他の制御情報です。DHCP クライアントは、要求されているオプションを判別して、DHCP パケットで送信します。

属性は名前と値のペアで、次のようなものがあります。

- DHCPv4 オプション。たとえば、**relay-agent-info**。
- DHCPv4　オプションから取得する情報のサブセット。たとえば、**relay-agent-remote-id**　は、DHCPv4 Option 82 のサブオプション 2 を表します。
- DHCPv4 オプションのフィールド。たとえば、「file」は DHCPv4 のヘッダーフィールドです。

属性には次のような設定も含めることもできます。

- Network Registrar の動作を制御する設定。たとえば、「drop」はパケットが破棄されることを示します。
- 情報を提供する設定。

BAC 4.0 と Network Registrar 7.0 では、次の 2 つの API バージョンをサポートします。BAC 拡張はそれらの API を使用して、DHCPv4 または DHCPv6 をイネーブルにします。

- DEX API バージョン 1：この API を使用すると、Network Registrar 拡張は、属性を介して DHCPv4 パケットの詳細をクエリーできます。
- DEX API バージョン 2：この API を使用すると、Network Registrar 拡張は、DHCPv4 オプションと DHCPv6 オプション、およびサブオプションに直接クエリーできます。

  BAC 拡張は、Network Registrar 拡張の 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と、DHCPv6 のサポートをイネーブルにします。

DHCPv6 用に検出されたデータを制御するプロパティ

BAC 拡張が DHCPv6 用に検出するデータを制御するプロパティには 3 種類のセットがあります。

（注） 管理者のユーザ インターフェイスを使用すると、**Configuration > Defaults > NR Defaults** ページでこれらのプロパティの設定を確認できます。

- バージョン 4.0 以前の Network Registrar 拡張の動作を制御するプロパティ。表 6-3 を参照してください。
- BAC 4.0 での DHCPv4 の Network Registrar 拡張の動作を制御するプロパティ。表 6-4 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、BAC 4.0 での DHCPv6 の Network Registrar 拡張の動作を制御するプロパティ。この違いは、DHCPv4 標準はクライアント メッセージとリレー メッセージを組み合せて 1 つのメッセージにし、それに対して、DHCPv6 標準はそれらのメッセージを分割するという点にあります。表 6-5 を参照してください。

表 6-3 は、BAC のバージョン 4.0 以前での Network Registrar 拡張の動作に影響を与えるプロパティについて説明しています。

表 6-3　BAC 4.0 以前の Network Registrar 拡張のプロパティ

プロパティ名	説明
/cnrExtension/attributesRequiredInRequest	RDU に構成生成の要求を送信するために、Network Registrar 要求辞書が拡張に含める必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_REQUIRED_IN_REQUEST_DICTIONARY`
/cnrExtension/attributesToPullFromRequestAsBytes	Network Request 要求辞書からバイナリ形式でプルする必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_REQUEST_DICTIONARY_AS_BYTES`
/cnrExtension/attributesToPullFromRequestAsStrings	Network Registrar 要求辞書から文字列形式でプルする必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_REQUEST_DICTIONARY_AS_STRINGS`
/cnrExtension/attributesToReadFromEnvironmentDictionary	Network Registrar 環境辞書からプルする必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_ENVIRONMENT_DICTIONARY`

表 6-4 では、BAC 4.0 での DHCPv4 の Network Registrar 拡張の動作を制御するプロパティについて説明しています。

表 6-4　BAC 4.0 の DHCPv4 Network Registrar 拡張のプロパティ

プロパティ名	説明
/cnrExtension/attributesRequiredInV4Request	RDU に構成生成の要求を送信するために、Network Registrar 要求辞書が拡張に含める必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_REQUIRED_IN_V4_REQUEST_DICTIONARY`
/cnrExtension/attributesToPullFromV4RequestAsBytes	Network Registrar 要求辞書からバイナリ形式でプルする属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_V4_REQUEST_DICTIONARY_AS_BYTES`
/cnrExtension/attributesToPullFromV4RequestAsStrings	Network Registrar 要求辞書から文字列形式でプルする属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_V4_REQUEST_DICTIONARY_AS_STRINGS`

表 6-5 では、BAC 4.0 での DHCPv6 の Network Registrar 拡張の動作を制御するプロパティについて説明しています。

表 6-5　BAC 4.0 の DHCPv6 Network Registrar 拡張のプロパティ

プロパティ名	説明
クライアント メッセージ	
/cnrExtension/attributesRequiredInV6 Request	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 要求辞書が拡張に含める必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_REQUIRED_IN_V6_REQUEST_DICTIONARY`
/cnrExtension/attributesToPullFromV6 RequestAsBytes	Network Registrar DHCPv6 要求辞書からバイナリ形式でプルする属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_V6_REQUEST_DICTIONARY_AS_BYTES`
/cnrExtension/optionsRequiredInV6 Request	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 要求辞書が拡張に含める必要がある DHCP オプションのリストを示します。
	API 定数 `CNR_OPTIONS_REQUIRED_IN_V6_REQUEST_DICTIONARY`
/cnrExtension/optionsToPullFromV6 Request AsBytes	Network Registrar DHCPv6 要求辞書からバイナリ形式でプルする DHCP オプションのリストを示します。
	API 定数 `CNR_OPTIONS_TO_READ_FROM_V6_REQUEST_DICTIONARY_AS_BYTES`
リレー メッセージ	
/cnrExtension/attributesRequiredInV6 Relay	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 Relay-Forward 要求辞書が拡張に含める必要がある属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_REQUIRED_IN_V6_RELAY_DICTIONARY`
/cnrExtension/attributesToPullFromV6 RelayAsBytes	Network Registrar DHCPv6 Relay-Forward 要求リレー辞書からバイナリ形式でプルする属性のリストを示します。
	API 定数 `CNR_ATTRIBUTES_TO_READ_FROM_V6_RELAY_DICTIONARY_AS_BYTES`
/cnrExtension/optionsRequiredInV6Relay	RDU に構成生成の要求を送信するために、Network Registrar DHCPv6 Relay-Forward 要求辞書が拡張に含める必要がある DHCP オプションのリストを示します。
	API 定数 `CNR_OPTIONS_REQUIRED_IN_V6_RELAY_DICTIONARY`
/cnrExtension/optionsToPullFromV6 RelayAsBytes	Network Registrar DHCPv6 Relay-Forward 要求リレー辞書からバイナリ形式でプルする DHCP オプションのリストを示します。
	API 定数 `CNR_OPTIONS_TO_READ_FROM_V6_RELAY_DICTIONARY_AS_BYTES`

IPv6 の設定ワークフロー

IPv6 をサポートするための BAC の設定には、2 つの異なるワークフローがあります。

- プロビジョニング グループでの DPE の設定。表 3-3 を参照してください。
- ネットワークでの Network Registrar サーバの設定。表 3-5 を参照してください。

リース クエリー

BAC RDU は、DHCP リース クエリー プロトコルを使用して、Network Registrar に対してデバイスの IP アドレスをクエリーします。その後、BAC はこの情報を IPv4 と IPv6 の両デバイスのデバイス中断や詳細なレポート作成に使用します。

この BAC リリースは、次の構成をサポートしています。

- リース クエリーの自動設定（P.6-19）
- リース クエリーの送信元 IP アドレス（P.6-19）

リース クエリーの自動設定

RDU は名前解決を実行して、リース クエリーの送信先である Network Registrar サーバの IP アドレスを決定します。DNS で障害が発生すると、リース クエリーは失敗します。この BAC リリースでは、RDU がリース クエリー要求を送信する必要がある、プロビジョニング グループ内の Network Registrar サーバの IP アドレスを直接設定できます。

自動設定をイネーブルにすると、RDU はそのリース クエリーの設定を調整して、プロビジョニング グループ内の Network Registrar サーバから IPv4 と IPv6 の両方のアドレス リストを設定します。RDU は、サーバに現在登録されている情報と RDU データベースに格納されている情報を比較してから、このタスクを実行します。BAC Network Registrar 拡張が 1 つのプロビジョニング グループから別のグループに移動している場合、リース クエリーの設定は変更され、次の内容が削除されます。

- 以前のプロビジョニング グループ オブジェクトに対するリース クエリーの設定に存在する IP アドレス。
- IP アドレス リストにもう存在しない IP アドレス。

RDU は、リース クエリーの設定を検索して、プロビジョニング グループが指定した拡張を使用するように設定されているかどうかを検証します。プロビジョニング グループが拡張を使用するように設定されていない場合、RDU は Network Registrar サーバに登録されているアドレスからアドレスを選択して、プロビジョニング グループのリース クエリーの設定に追加します。

この自動設定をディセーブルにした場合、RDU は Network Registrar サーバでの登録時にそのリース クエリーの設定を変更しません。デフォルトでは、この機能はイネーブルになっています。

プロビジョニング グループ内のリース クエリー アドレスの自動設定をイネーブルまたはディセーブルにするには、管理者ユーザ インターフェイスから LeaseQuery AutoConfig オプションを設定できます。P.12-29 の「プロビジョニング グループの表示」を参照してください。

リース クエリーの送信元 IP アドレス

以前のバージョンの BAC では、リース クエリー機能は、リース クエリー要求を送信するための送信元インターフェイスと送信元ポートの選択をオペレーティング システムに依存していました。このリリースではこれがデフォルトの動作ですが、特定のインターフェイスを使用してリース クエリー要求を送信するように RDU を設定することもできます。

リース クエリーの設定

デフォルトでは、BAC は表 6-1 に記載されている IP アドレスとポートにバインドします。

表 6-1　バインディング用のリース クエリー アドレス

プロトコル	IP アドレス	ポート
IPv4	ワイルドカード[1]	67
IPv6	ワイルドカード	547

1. ワイルドカードは、特別なローカル IP アドレスです。通常は「すべて」を意味し、バインド操作でのみ使用できます。

RDU でポート 547 とポート 67 が利用可能な場合は、リース クエリー要求を送信するために特別な設定を行う必要はありません。RDU のインストール中に、インストール プログラムがこれらのポートのいずれかが他のプロセスで使用されていることを検出した場合、オペレーティング システムが選択する動的ポートを使用することをお勧めします。

次に例を示します。

```
DHCPv4/DHCPv6 lease query port(s) (Udp/67 and Udp/547) is in use.
Configuring the RDU to use a dynamic port for DHCPv4/DHCPv6 lease query.
```

インストール プログラムは、*BPR_HOME/rdu/conf/rdu.properties* ファイルの次のプロパティに値 0 を設定して動的ポートの選択を自動的にイネーブルにします。

```
/cnrQuery/clientSocketAddress=0.0.0.0:0
/cnrQuery/ipv6/clientSocketAddress=[::]:0
```

同じプロパティを使用して、リース クエリーの通信に使用する IP アドレスとポートを設定することもできます。次に例を示します。

```
/cnrQuery/clientSocketAddress=10.1.2.3:166
/cnrQuery/ipv6/clientSocketAddress=[2001:0DB8:0:0:203:baff:fe12:d5ea]:1547
```

RDU は、これらのプロパティを使用して、ユーザが指定する IP アドレスとポートにバインドします。

（注） *rdu.properties* ファイルのプロパティを手動で変更する場合は、必ず RDU を再起動してください。**/etc/init.d/bprAgent restart rdu** コマンドを使用します。

リース クエリーのリレー エージェントとしての BAC の設定

リレー エージェントとして動作するように BAC を設定できます。リレー エージェントのオプションは次のとおりです。

- IPv4 ではデフォルトでイネーブル化
- IPv6 ではデフォルトでディセーブル化

IPv4 のリース クエリーの場合

BAC が IPv4 リース クエリーのリレー エージェントとして動作する場合、BAC はリース クエリー要求パケットに GIADDR（DHCP サーバが応答する IP アドレス）を提供します。デフォルトでは、RDU は、この目的でコンピュータのプライマリ IP アドレスを使用します。

(注) 配備内のすべての DHCP サーバがこの IP アドレスにアクセスできるようにしてください。また、このプロパティで使用する IP アドレスは、RDU をインストールしたコンピュータに存在する必要があります。

GIADDR フィールドで使用する IP アドレスを変更するには、*rdu.properties* ファイルの */cnrQuery/giaddr* プロパティの値を変更する必要があります。たとえば、GIADDR を 10.10.10.1 に変更する場合、次のように追加します。

```
/cnrQuery/giaddr=10.10.10.1
```

rdu.properties ファイルのプロパティを手動で変更する場合は、**/etc/init.d/bprAgent restart rdu** コマンドを使用して RDU を再起動してください。

IPv6 リース クエリーの場合

BAC が IPv6 リース クエリーのリレー エージェントとして動作するように設定するには、*rdu.properties* ファイルに次のプロパティを含める必要があります。

/cnrQuery/ipv6/linkAddress=IPv6 address

/cnrQuery/ipv6/peerAddress=IPv6 address

次に例を示します。

```
/cnrQuery/ipv6/linkAddress=2001:0DB8:0:0:203:baff:fe12:d5ea
/cnrQuery/ipv6/peerAddress=2001:0DB8:0:0:203:baff:fe12:d5ea
```

(注) リンク アドレスとピア アドレスに入力する値は、BAC と Network Registrar を稼動するネットワークの構成によって異なります。単純なケースでは、リンク アドレスとピア アドレスを RDU ホストの IPv6 アドレスに設定する必要があります。この IPv6 アドレスは Network Registrar にルーティング可能である必要があります。

/etc/init.d/bprAgent restart rdu コマンドを使用して、RDU を再起動します。

例

この例は、リレー エージェント オプションがイネーブルになっている IPv6 リース クエリー要求の出力を示しています。

```
rdu.example.com: 2007 10 18 19:40:30 EDT: %BAC-RDU-7-DEBUG_DHCP_IF_IPV6:
PACE-2:ServerBatch[Batch:rdu.example.com/10.10.10.1:1b994de:115b52abeb4:80000278]:
Peer[rdu.example.com:33743]: Querying single prov group for DUID
[00:03:00:01:23:45:67:89:98:56] via DHCPv6 LEASEQUERY packet [version V6, message-type
12, hop-count 0, link-address 2001:0DB8:0:0:203:baff:fe12:d5ea, peer-address
2001:0DB8:0:0:203:baff:fe12:d5ea, (relay_msg (9) option (52 bytes) version V6,
message-type 14, transaction-id 13401290, (client-identifier (1) option (9 bytes)
00:11:22:33:44:55:66:77:88), (lq-query (44) option (31 bytes) query-type 2,
link-address 0:0:0:0:0:0:0:0, (client-identifier (1) option (10 bytes)
00:03:00:01:23:45:67:89:98:56)))]
```

AIC Echo のイネーブル化

AIC Echo オプションを使用して、標準ポートではなく、要求元のクライアントの送信元ポートに応答を送信するように Network Registrar を設定できます。

たとえば、IP アドレスが 10.1.1.1 のクライアントがポート 1456 を使用して要求を転送し、サーバ上で AIC Echo がディセーブルになっている場合、そのサーバは応答を標準クライアント ポートに返します。プロトコル スタックに応じて、標準クライアント ポートは次のいずれかになります。

- 67 は IPv4
- 546 は IPv6

AIC Echo がイネーブルになっている場合、応答はポート 1456 に転送されます。

IPv4 リース クエリーを要求する場合、AIC Echo はデフォルトでディセーブルになります。このオプションは、デフォルトの IPv4 バインディング ポートが変更された場合にのみ使用されます。

IPv6 リース クエリーを要求する場合、AIC Echo はデフォルトでイネーブルになります。ただし、IPv6 リース クエリー メッセージはデフォルトではリレーされないので、このオプションは、標準クライアント ポートの 546 ではなく、ポート 547 にリース クエリーの応答を返すために使用されます。

リース クエリーのデバッグ

RDU で情報レベル ロギング（6：情報）を使用して、リース クエリー処理に関連する重要な詳細情報を表示できます（RDU でのログ レベルを設定するには、P.10-5 の「RDU ログ レベル ツールの使用方法」を参照してください）。

リース クエリー機能をデバッグする場合は、次のプロパティを使用できます。

- *dhcpleasequeryv4*：IPv4 リース クエリーのデバッグ
- *dhcpleasequeryv6*：IPv6 リース クエリーのデバッグ

IPv6 リース クエリーの使用例

この項では、次の IPv6 リース クエリーの使用例について説明します。

- プロビジョニング グループ内の全（2 台）Network Registrar サーバにおけるクライアントごとに 1 つのリース
- プロビジョニング グループ内の全（2 台）Network Registrar サーバにおけるクライアントごとに複数のリース
- 1 台の Network Registrar サーバのクライアントごとに複数のリース
- 委任プレフィックスがあるデバイスのリース

プロビジョニング グループ内の全（2 台）Network Registrar サーバにおけるクライアントごとに 1 つのリース

IPv6 に対するフェールオーバー プロトコルがまだ定義されていない状態では、通常、プロビジョニング グループ内の 1 台の Network Registrar サーバのみがクライアントのリース情報を保持します。このケースで、プロビジョニング グループに 2 台の Network Registrar サーバが存在する場合、RDU は両方のサーバにリース クエリー要求を送信しますが、1 台からのみ応答を受信します。IP アドレスはその応答で指定されたものが使用されます。

この IP アドレスは、次の方法で表示できます。

- 管理者のユーザ インターフェイス（**Devices** > **Device Details** ページ上）。
- API（リース クエリー マップで `client-ipaddress` 属性を使用）。

プロビジョニング グループ内の全（2 台）Network Registrar サーバにおけるクライアントごとに複数のリース

まれに、プロビジョニング グループ内の両方の Network Registrar サーバが同じクライアントのリースを保持し、両方のサーバがリース クエリー応答に応答することがあります。この場合、DHCPv6 Leasequery ドラフトごとに、直近の OPTION_CLT_TIME（client-last-transaction-time）がある応答が使用されます。

1 台の Network Registrar サーバのクライアントごとに複数のリース

クライアントが同じサーバの異なる 2 つのリンクにリースを保持している場合、Network Registrar では、応答時の OPTION_LQ_CLIENT_LINK オプションにすべてのリンク アドレスが含まれます。その後、BAC は Network Registrar に個々のリンクをクエリーして、すべての IP アドレスを取得します。このリストでは、BAC は、ループバックでもデバイス中断用のマルチキャスト アドレスでもない最初の IP アドレスを使用します。

管理者のユーザ インターフェイスを使用して、このプロセスで取得した IP アドレスのリストを **Devices** > **Device Details** ページで表示できます。

委任プレフィックスがあるデバイスのリース

IP アドレス、または委任プレフィックス、あるいは両方が割り当てられているデバイスのリース クエリー要求を送信できます。

管理者のユーザ インターフェイスを使用して、IP アドレスとプレフィックスを **Devices** > **Device Details** ページで表示できます。この IP アドレスを API を使用して取得するには、リース クエリー マップで `iaprefix` 属性を使用します。

CHAPTER 7

PacketCable 音声設定

この章では、PacketCable 音声を配備して使用するために実行する必要がある作業について説明します。

この章は、PacketCable の次の使用形態に関する情報を記載しています。

- PacketCable eMTA のセキュア プロビジョニング（P.7-2）
- PacketCable eMTA の Basic プロビジョニング（P.7-31）
- Euro PacketCable（P.7-33）

PacketCable 音声技術の配備における問題を解決するための情報については、P.16-12 の「PacketCable eMTA プロビジョニングのトラブルシューティング」を参照してください。

この章は、PacketCable Multimedia Terminal Adapter（MTA; マルチメディア ターミナル アダプタ）デバイスのプロビジョニング仕様、PKT-SP-PROV1.5-I03-070412 の内容を熟知している読者を対象としています。詳細については、PacketCable の Web サイトを参照してください。

PacketCable eMTA のセキュア プロビジョニング

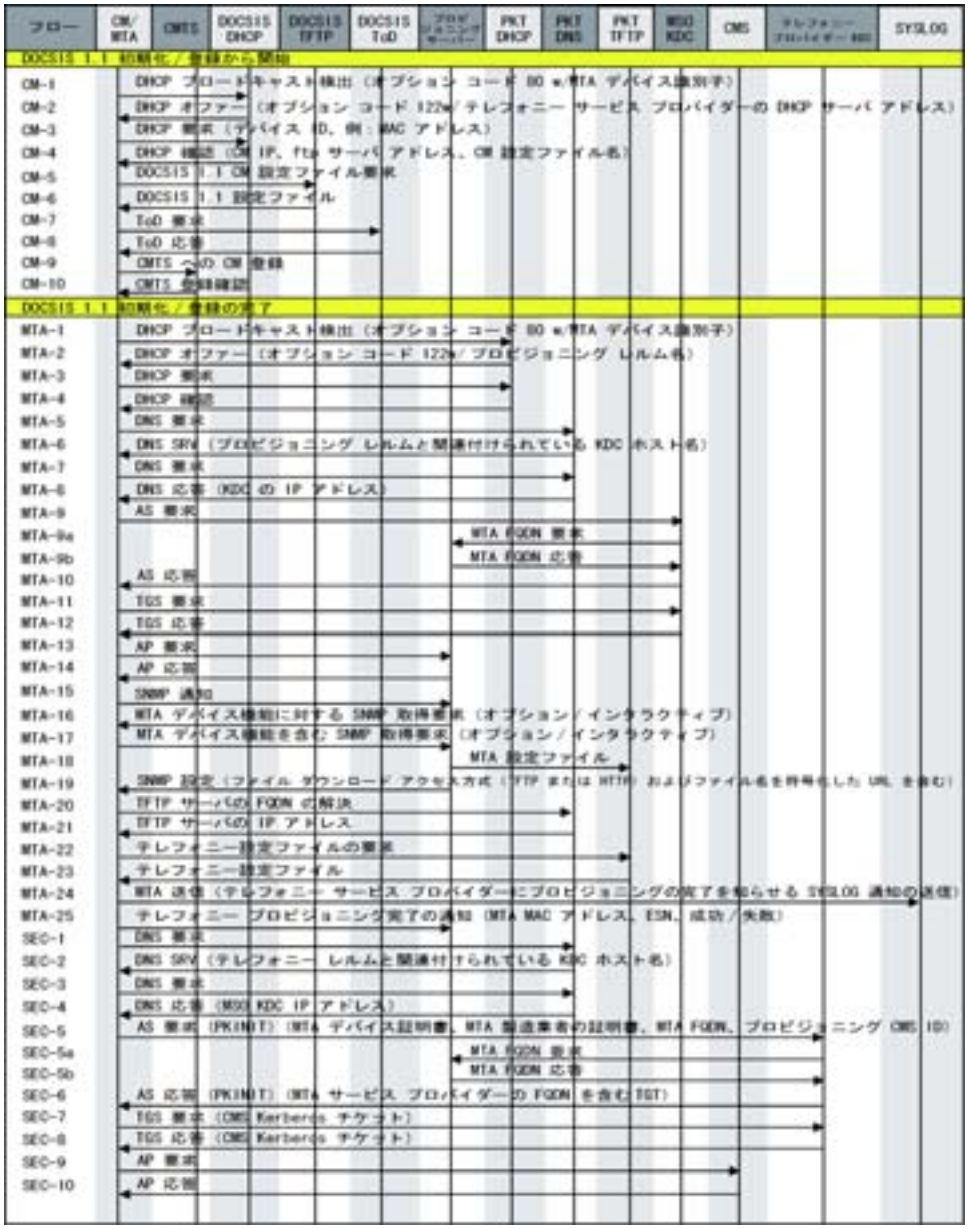
この項では、主にセキュアな PacketCable 音声プロビジョニングについて取り上げます。PacketCable Secure は、テレフォニー サービスの盗用や、悪意のあるサービス中断などが発生する可能性を最小限に抑えるように設計されています。PacketCable Secure では、Kerberos インフラストラクチャを利用して、MTA とプロビジョニング システムを相互に認証します。BAC では、Key Distribution Center（KDC; 鍵発行局）は Kerberos サーバとして機能します。MTA とプロビジョニング システム間の対話をセキュリティで保護するために、SNMPv3 も使用されます。

BAC PacketCable のセキュアなプロビジョニングのフロー

PacketCable プロビジョニングのすべてのフローは、一連のステップとして定義されます。

図 7-1 に、PacketCable eMTA のセキュアなプロビジョニングのフローを示します。

図 7-1　組み込み型 MTA のセキュアなパワーオン プロビジョニング フロー



（注）データ パケットをキャプチャできるプロトコル アナライザ（プロトコル スニファ）を使用して、どのステップが失敗しているかを正確に把握することを強くお勧めします。

また、KDC の障害の根本的な原因を把握するには、KDC ログ ファイルの内容が重要です。

embedded Multimedia Terminal Adapter（eMTA; 組み込み型マルチメディア ターミナル アダプタ）のプロビジョニングにおける問題を診断する場合、表 7-1 のフローの説明が、PacketCable の失敗しているプロビジョニング フローのステップを特定するのに役立ちます。

表 7-1　PacketCable eMTA のセキュア プロビジョニング

手順	ワークフロー	説明
CM-1	DHCP ブロードキャスト検出	これは、DHCPv4 または DHCPv6 の DOCSIS ケーブル モデム（CM）ブートフローと同様で、PacketCable DHCP サーバのリストを MTA に提供するための DHCP オプションが付加されます。MTA はこれらの DHCP サーバから DHCP オファーを受け付けられるようになります。
CM-2	DHCP オファー	
CM-3	DHCP 要求	
CM-4	DHCP 確認	
CM-5	DOCSIS 1.1 CM 設定ファイル要求	
CM-6	DOCSIS 1.1 設定ファイル	
CM-7	ToD 要求	
CM-8	ToD 応答	
CM-9	CMTS（ケーブル モデム ターミネーション システム）への CM の登録	
CM-10	CMTS 登録確認応答	

表 7-1　PacketCable eMTA のセキュア プロビジョニング（続き）

手順	ワークフロー	説明
MTA-1	DHCP ブロードキャスト検出	DHCP を使用して、MTA が自身を PacketCable MTA としてアナウンスし、サポートしている機能およびプロビジョニング フロー（Secure、Basic など）についての情報を提供します。MTA はアドレス解決情報と DHCP Option 122 も取得します。DHCP Option 122 は、PacketCable のプロビジョニング サーバのアドレスとセキュリティ レルム名を含んでいます。この情報は、MTA から KDC およびプロビジョニング サーバへのアクセスを可能にするために使用されます。 基本的なトラブルシューティングのヒントを次に示します。 • CMTS 上で DHCP リレー エージェントが正しく設定されていることを確認する。CMTS が正しい DHCP サーバをポイントしていることを確認します。 • MTA、CMTS、DHCP サーバ、DPE 間のルーティングが正しいことを確認する。 • セカンダリ サブネットが CMTS 上で正しく設定されていることを確認する。 • Cisco Network Registrar の DHCP 設定が正しいことを確認する。スコープが設定され、IP アドレスが利用可能であり、すべてのセカンダリ サブネットが設定されていることを確認します。 • BAC の設定を確認する。*cnr_ep.properties* ファイルを確認して、必要な PacketCable Network Registrar 拡張のプロパティが設定されていることを確認します。詳細については、付録 C「PacketCable DHCP オプションと BAC プロパティのマッピング」を参照してください。 MTA がフロー ステップの MTA-1 と MTA-2 間で循環していることをパケット トレースが発見した場合は、DHCP Option 122（レルム名またはプロビジョニング サーバ FQDN のサブオプション）、DHCP Option 12（ホスト名）、または DHCP Option 15（ドメイン名）の設定に問題がある可能性があります。
MTA-2	DHCP オファー	
MTA-3	DHCP 要求	
MTA-4	DHCP 確認	
MTA-5	DNS 要求	MTA が（DHCP Option 122 で提供される）セキュリティ レルム名を使用して、KDC サービスに対して DNS SRV ルックアップを実行し、KDC の IP アドレスを解決します。 基本的なトラブルシューティングのヒントを次に示します。 • パケット スニファを使用して、Network Registrar DNS に送信される、送信先や形式が不正な DNS パケットを検出する。 • Network Registrar DNS のログ レベルをパケット詳細トレースに設定して、Network Registrar DNS にどのようなパケットが到達するかを確認する。 • DNS 設定を確認する：*cnr_ep.properties* に指定されている DNS サーバは、KDC のレルム ゾーン、SRV レコード、および DNS「A」レコードを保持している必要があります。
MTA-6	DNS Srv	
MTA-7	DNS 要求	
MTA-8	DNS 応答	
MTA-9	AS 要求	AS-REQ 要求メッセージが KDC によって使用され、MTA が認証されます。 基本的なトラブルシューティングのヒントを次に示します。 • KDC のログ ファイルを確認して、AS-REQ が到達しているかどうかを判断し、エラーや警告がないことを確認する。 • KDC が正しい MTA_Root 証明書を使用して設定されていることを確認する。MTA が AS-REQ メッセージに添付して送信する製造業者証明書およびデバイス証明書は、KDC にインストールされている MTA_Root 証明書とチェーンを構成する必要があります。

表 7-1　PacketCable eMTA のセキュア プロビジョニング（続き）

手順	ワークフロー	説明
MTA-9a	MTA FQDN 要求	KDC が MTA の MAC アドレスを MTA 証明書から抽出して、検証のためにプロビジョニング サーバに送信します。プロビジョニング サーバがこの MAC アドレスの FQDN を保持している場合は、FQDN が KDC に返されます。KDC は MTA から受信した FQDN を FQDN-REP 応答メッセージで受信した FQDN と比較します。 基本的なトラブルシューティングのヒントを次に示します。 • パケット スニファを使用して、送信先や形式が不正な DNS パケットを検出する。MTA は、（MTA が DHCP Option 122 で受信した）プロビジョニング サーバの FQDN を AS-REP メッセージ内で KDC に渡します。KDC はこの FQDN を使用して、プロビジョニング サーバの IP アドレスを解決します。 • KDC キー ファイルのファイル名と内容を確認する。DPE 内の KDC サービス キーは、KDC にあるサービス キーと一致している必要があります。KDC にあるサービス キー ファイルの名前は、非常に重要です。
MTA-9b	MTA FQDN 応答	
MTA-10	AS 応答（AS-REP）	KDC がプロビジョニング サービス チケットを MTA に付与し、サービス プロバイダー証明書、ローカル システム プロバイダー証明書（オプション）、および KDC 証明書を MTA に送信します。MTA は KDC から送信された証明書が、MTA に格納されているサービス プロバイダーのルート証明書とチェーンを構成していることを確認します。これらの証明書がチェーンを構成していない場合、MTA はプロビジョニング フローのステップ MTA-1 に処理を戻します。*KDC.cer* ファイルの詳細については、P.14-3 の「PKCert.sh ツールの使用方法」を参照してください。 基本的なトラブルシューティングのヒントを次に示します。KDC のログ ファイルを表示して、AS-REP メッセージがデバイスに送信されたことを確認する。MTA がステップ MTA-1 ～ MTA-10 を循環していることをパケット トレースが発見した場合は、サービス プロバイダー証明書チェーンに問題があります。
MTA-11	TGS 要求	ステップ MTA-10 の後、MTA がサービス チケットまたは Ticket-Granting-Ticket（TGT; チケット認可チケット）を受信します。MTA はステップ MTA-10 で、サービス チケットの代わりに TGT を取得した場合、Ticket-Granting-Server（KDC）にアクセスして、サービス チケットを取得します。
MTA-12	TGS 応答	KDC が TGS 応答内のサービス チケットを MTA に送信します。
MTA-13	AP 要求（AP-REQ）	MTA が（ステップ MTA-10 で受信した）チケットを DHCP Option 122 で指定されているプロビジョニング サーバに提示します。
MTA-14	AP 応答（AP-REP）	プロビジョニング サーバが KDC 共有秘密情報を使用して AP-REQ を復号化し、MTA が提示したプロビジョニング サーバ チケットを検証して、AP-REP を SNMPv3 キーを使用して送信します。以後の SNMPv3 は認証済みになり、必要に応じて暗号化されます。
MTA-15	SNMP 通知	MTA が、プロビジョニング情報を受信可能なことをプロビジョニング サーバに通知します。
MTA-16	SNMP 取得要求	SNMPv3：プロビジョニング サーバ（DPE）が追加のデバイス機能を必要とする場合は、MTA に 1 つ以上の SNMPv3 取得要求を送信して、MTA 機能に関する必要な情報を取得します。プロビジョニング サーバ（DPE）は、一括取得要求を使用して、1 つのメッセージに大量の情報を要求することがあります。

表 7-1　PacketCable eMTA のセキュア プロビジョニング（続き）

手順	ワークフロー	説明
MTA-17	SNMP 取得応答	SNMPv3：MTA が、ステップ MTA-16 で要求した MTA 機能に関する情報を含む各取得要求に対する応答をプロビジョニング サーバ（DPE）に送信します。
MTA-18	MTA 設定ファイル	ステップ MTA-16 と MTA-17 で利用可能にした情報を使用して、プロビジョニング サーバ（DPE）が MTA 設定データ ファイルの内容を判別します。
MTA-19	SNMP 設定	SNMPv3：プロビジョニング サーバが、MTA 設定ファイルの URL、このファイルの暗号キー、およびこのファイルのハッシュ値を含んだ SNMPv3 設定を MTA に対して実行します。
MTA-20	TFTP サーバの FQDN の解決	DNS 要求：URL 符号化アクセス方式に IPv4 アドレスの代わりに FQDN が含まれている場合、MTA はサービス プロバイダー ネットワークの DNS サーバを使用して、FQDN を解決して TFTP サーバまたは HTTP サーバの IPv4 アドレスにします。
MTA-21	TFTP サーバの IP アドレス	DNS 応答：DNS サーバが、ステップ MTA-20 で要求したサービス プロバイダー ネットワークの IPv4 IP アドレスを返します。
MTA-22	テレフォニー設定ファイル要求	MTA は、指定された TFTP サーバから VoIP 設定ファイルをダウンロードする処理に進みます。BAC は、TFTP サーバを DPE コンポーネントに統合します。
MTA-23	テレフォニー設定ファイル	
MTA-24	MTA 送信	MTA がオプションで、プロビジョニングの完了を知らせる syslog 通知をサービス プロバイダーに送信します。
MTA-25	テレフォニー プロビジョニングの完了通知	MTA が、新しい設定の受け付けが可能かどうかをプロビジョニング サーバに通知します。
SEC-1 ～ SEC-10	これらのステップは、ポスト MTA プロビジョニングのセキュリティ フローのもので、BAC プロビジョニングには適用できません。このフローには、MTA が通信する各 CMS に関連付けられている Kerberos チケットの取得が含まれます。詳細については、PacketCable Security の仕様を参照してください。	

PacketCable eMTA のセキュア プロビジョニングにおける KDC

PacketCable Secure では、Kerberos インフラストラクチャを利用して、MTA とプロビジョニング システムを相互に認証します。BAC 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、P.2-14 の「Key Distribution Center」を参照してください。

KDC に関する重要な情報については、次を参照してください。

- KDC のデフォルト プロパティ（P.7-7）
- KDC 証明書（P.7-9）
- KDC ライセンス（P.7-9）
- 複数レルムのサポート（P.7-10）

KDC のデフォルト プロパティ

KDC には、BAC インストール中に *BPR_HOME/kdc/solaris/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、操作要件で指示された値に変更することができます。

（注） 動作要件を記述する場合は、*kdc.ini* ファイルの編集には注意してください。誤った値を指定すると KDC が動作しなくなる場合があります。変更を加えた場合は、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 が着信 Kerberos メッセージを監視するローカルのイーサネット 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ているコンピュータの Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を示します。

  次に例を示します。

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

（注） インターフェイス アドレスと FQDN 値は、インストール時に KDC Realm Name 画面から入力する必要があります。具体的な情報については、『*Installation and Setup Guide for Cisco Broadband Access Center 4.0*』を参照してください。

- maximum log file size：KDC が生成するログ ファイルの拡大可能な最大サイズを KB 単位で指定します。KDC は、現在のファイルがこの最大サイズに達した場合にのみ新しいログ 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。必要な数だけ指定できます。

  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
  ```

  表 7-2 では、KDC のログ ファイルで使用可能なロギング レベルについて説明します。

表 7-2　KDC のロギング レベル

ログ レベル	説明
0	エラー状態が存在します。すべてのエラー メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
1	警告状態が存在します。すべての警告メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
2	情報メッセージ。利用可能なすべてのロギング メッセージを保存するように、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 までの、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ように、ロギング機能を設定します。

- minimum (maximum) ps backoff：KDC が FQDN-Request に応答するために BAC を待つ最短（または最長）待ち時間を 0.1 秒単位で指定します。

  次に例を示します。

  ```
  minimum ps backoff = 150
  ```

上述のサンプル値を使用する、サンプル INI ファイルには、例 7-1 に表示されているのと同様のデータが含まれます。

例 7-1　kdc.ini 設定ファイルのサンプル

```
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
```

配備中に発生する可能性がある多数のチケット要求を効率的に処理するために、最短と最長のチケット継続期間のそれぞれに時間を設定できます。この設定は、ほとんどの配備が通常の業務時間内に行われ、ときどき過剰な負荷がかかり、パフォーマンスに悪影響を与える場合に有益です。

（注） チケット期間を短縮すると、MTA はより頻繁に KDC を認証するようになります。この結果、テレフォニー エンドポイントの認可をさらに管理できるようになりますが、KDC でのメッセージ負荷が増し、ネットワーク トラフィックが増えることにもなります。ほとんどの場合はデフォルトの設定が適しているため、変更しないでください。

- maximum ticket duration：KDC が生成するチケットの最長継続期間を定義します。デフォルトの単位は時間ですが、**m** または **d** を追加すると、単位をそれぞれ分または日に変更できます。

  デフォルト値は 168（7 日）です。この値は、PacketCable Security の仕様を確認するために必要な時間の長さであるため、この値は変更しないことをお勧めします。

  次に例を示します。

  ```
  maximum ticket duration = 168
  ```

- minimum ticket duration：KDC が生成するチケットの最短継続期間を定義します。デフォルトの単位は時間ですが、**m** または **d** を追加すると、単位をそれぞれ分または日に変更できます。

  デフォルト値は 144（6 日間）です。この値は変更しないことをお勧めします。

  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は、BAC に同梱されていません。Cable Television Laboratories, Inc.（CableLabs）から必要な証明書を入手する必要があります。これらの証明書の内容は、MTA にインストールされている証明書の内容と一致している必要があります。

（注） 証明書がインストールされていないと KDC は機能しません。

PKCert ツールを使用して、KDC が動作するために必要な証明書をインストールして、管理することができます。PKCert ツールは、CableLabs サービス プロバイダー証明書を証明書ファイルとしてインストールします。このツールの実行方法の詳細については、P.14-3 の「PKCert.sh ツールの使用方法」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のみ利用できます。

KDC ライセンス

シスコの代理店から KDC ライセンスを入手して、正しいディレクトリにインストールしてください。

KDC ライセンス ファイルをインストールするには、次の手順に従います。

ステップ 1　シスコの代理店からライセンス ファイルを入手します。

ステップ 2　BAC ホストに *root* としてログインします。

ステップ 3　ライセンス ファイルをコピーします。

注意　ファイルを ASCII ファイルとしてコピーしないように注意してください。このファイルには、ASCII 転送中に不要な変更が加えられやすいバイナリ データが含まれています。

KDC ライセンス ファイルは転送プロセスにより損傷を受ける可能性があるため、異なるオペレーティング システム間でコピーすることはできません。

ステップ 4　KDC サーバを再起動して、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。

複数レルムのサポート

BAC KDC は複数レルムの管理をサポートします。この場合、有効な PacketCable X.509 証明書と KDC 秘密鍵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/solaris/packetcable/certificates* ディレクトリに存在する必要があります。

BAC は、*BPR_HOME/kdc/solaris/packetcable/certificates* ディレクトリの下にサブディレクトリをインストールすることで、追加のレルムをサポートします。各サブディレクトリには、特定のレルムと同じ名前が付けられます。

表 7-3 に、さまざまな証明書とそれぞれに対応するファイル名を示します。これらのファイルは、*BPR_HOME/kdc/solaris/packetcable/certificates* ディレクトリに存在する必要があります。

表 7-3　PacketCable の証明書

証明書	証明書のファイル名
MTA のルート	*MTA_Root.cer*
サービス プロバイダーのルート	*CableLabs_Service_Provider_Root.cer*
サービス プロバイダーの CA	*Service_Provider.cer*
ローカル システム オペレータの CA	*Local_System.cer*
KDC	*KDC.cer*

プライマリ レルムは、KDC コンポーネントのインストール中に設定されます。プライマリ レルムの場合、KDC 証明書（*KDC.cer*）は、*BPR_HOME/kdc/solaris/packetcable/certificates* ディレクトリに存在します。その秘密鍵（KDC_private_key.pkcs8）は、*BPR_HOME/kdc/solaris/* ディレクトリにあります。

追加のレルムを設定するには、次の手順に従ってください。この後で詳細を説明します。

ステップ 1　KDC 証明書を含んでいるディレクトリを検索します。

ステップ 2　そのディレクトリの下に KDC 証明書を格納するサブディレクトリを作成します。

（注）　サブディレクトリの名前を特定のレルムの名前と一致させます。大文字のみを使用して、サブディレクトリの名前を付けます。

ステップ 3　作成したサブディレクトリに、レルムの KDC 証明書と秘密鍵を配置します。

ステップ 4　新しいレルムが KDC 証明書として同じサービス プロバイダーとチェーンを構成していない場合は、証明書ディレクトリにあるものとは異なる追加の上位レベル証明書をすべて含めます。

（注）　すべてのレルムは同じ証明書チェーンにルートがある必要があるため、KDC のインストールは一度に 1 つのロケール（北米版 PacketCable または欧州版 PacketCable）のみサポートします。

表 7-4 では、プライマリ レルム（CISCO.COM など）と 2 つのセカンダリ レルム（CISCO1.COM と CISCO2.COM など）のディレクトリ構造とファイルについて説明します。この構造は、上位レベルの証明書がプライマリ レルムとそのセカンダリ レルムで同様であることを想定しています。

表 7-4　複数レルムのディレクトリ構造

ディレクトリ	ディレクトリのファイルの内容
BPR_HOME/kdc/solaris	プライマリ レルム CISCO.COM の場合： KDC 秘密鍵
BPR_HOME/kdc/solaris/packetcable/certificates	プライマリ レルム CISCO.COM の場合： • *MTA_Root.cer* • *CableLabs_Service_Provider_Root.cer* • *Service_Provider.cer* • *Local_System.cer* • *KDC.cer* Directory /*CISCO1.COM* Directory /*CISCO2.COM*
BPR_HOME/kdc/solaris/packetcable/certificates/CISCO1.COM	セカンダリ レルム CISCO1.COM の場合： • *KDC.cer* • KDC 秘密鍵
BPR_HOME/kdc/solaris/packetcable/certificates/CISCO2.COM	セカンダリ レルム CISCO2.COM の場合： • *KDC.cer* • KDC 秘密鍵

複数レルムの KDC の設定

この項では、複数レルムの KDC を設定するためのワークフローについて説明します。処理を進める前に、RDU、DPE、および Network Registrar 拡張のインストールを完了してください。インストールの説明については、『*Installation and Setup Guide for the Cisco Broadband Access Center 4.0*』を参照してください。

次のワークフローでは、サンプルのレルムとディレクトリを使用して、複数レルムの KDC を設定する方法を説明します。ここで使用するプライマリ レルムは CISCO.COM、そのセカンダリ レルムは CISCO1.COM と CISCO2.COM です。

次のワークフローのセットアップでは、3 つの MTA（Motorola SBV 5120 MTA、Linksys CM2P2 MTA、および SA WebStar DPX 2203 MTA）をプロビジョニングします。各 MTA は 1 つのレルムでプロビジョニングされます。Motorola は CISCO.COM レルム、Linksys MTA は CISCO1.COM レルム、SA MTA は CISCO2.COM レルムです。

（注） 次の手順で示されている出力例は、説明を目的としているため短く編集されています。

複数レルムの KDC を設定するには、次の手順に従います。

ステップ 1 DPE 上の次の設定内容を確認します。

a. **show run** コマンドを使用して、PacketCable サービスがイネーブルになっていることを確認します。

PacketCable サービスをイネーブルにするには、**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 local
  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 Broadband Access Center DPE CLI Reference 4.0*』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が設定されていることを確認します。

セキュリティ鍵を生成して設定するには、**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 local
  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 Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

c. PacketCable SNMPv3 クローニングに対する DPE と RDU 間での安全な通信を許可するセキュリティ鍵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。セキュリティ鍵を生成して設定するには、**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 local
  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>
```

コマンドの詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

（注） DPE 上で PacketCable を設定する場合は、**dpe reload** コマンドを実行して変更を有効にしてください。

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（この場合は、CISCO.COM）で設定されているかどうかを確認します。確認を行うには、*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.
#This file was created on Wed, March 4 06:34:34 EDT 2007
/rdu/port=49187
/rdu/fqdn=dpe4.cisco.com
/cache/provGroupList=Default
/cnr/sharedSecret=fggTaLg0XwKRs
/pktcbl/enable=enabled
/ccc/tgt=01
/ccc/kerb/realm=CISCO.COM
/ccc/dhcp/primary=10.10.0.1
/ccc/dns/primary=10.10.0.1
```

ステップ 3 静的ルートを適切にイネーブルにして、BAC と CMTS の背後にあるデバイスとの接続を確保します。

ステップ 4 *cnr_ep.properties* ファイルにリストされている DNS サーバの DNS レルム ゾーンを作成します。ゾーンは、**DNS > Forward Zones > List/Add Zones** ページから Network Registrar の管理者のユーザ インターフェイスを使用して追加できます。

（注） 追加するゾーンには、KDC サーバの SRV レコードと DNS「A」レコードを含め、各ゾーン（この場合は、CISCO.COM、CISCO1.COM、および CISCO2.COM）の SRV レコードが 1 つの KDC をポイントするようにしてください。

管理者のユーザ インターフェイスを使用してゾーンを設定する方法については、『*User Guide for Cisco Network Registrar 7.0*』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. セカンダリ レルム（この場合は、CISCO1.COM と CISCO2.COM）のディレクトリを *BPR_HOME/kdc/solaris/packetcable/certificates* の下に作成します。

次に例を示します。

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

ディレクトリの作成方法の詳細については、Solaris 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する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されている */certsInput* ディレクトリは、一例です。他のディレクトリの下に任意の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方法の詳細については、Solaris のマニュアルを参照してください。

c. 前のステップで示した証明書を、作成したディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

d. 次の証明書を *BPR_HOME/kdc/solaris/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*

ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

e. プライマリ レルムの KDC 証明書とそれに関連付けられる秘密鍵を作成します。

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

ツールの詳細については、P.14-3 の「PKCert.sh ツールの使用方法」を参照してください。

f. *KDC.cer* ファイルを KDC の証明書ディレクトリ（*BPR_HOME/kdc/solaris/packetcable/certificates*）にコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

g. 秘密鍵 KDC_private_key.pkcs8 を KDC のプラットフォーム ディレクトリ（*BPR_HOME/kdc/solaris*）にコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

h. 秘密鍵 KDC_private_key_proprietary. を KDC のプラットフォーム ディレクトリ（*BPR_HOME/kdc/solaris*）にコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

i. セカンダリ レルム（この例では、CISCO1.COM）の KDC 証明書とそれに関連付けられる秘密鍵を作成します。

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

ツールの詳細については、P.14-3 の「PKCert.sh ツールの使用方法」を参照してください。

j. *KDC.cer* をセカンダリ レルムのディレクトリにコピーします。たとえば、*BPR_HOME/kdc/solaris/packetcable/certificates* の下の */CISCO1.COM* ディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

k. 秘密鍵 KDC_private_key.pkcs8 をセカンダリ レルムのディレクトリにコピーします。たとえば、*BPR_HOME/kdc/solaris/packetcable/certificates* の下の */CISCO1.COM* ディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

l. 秘密鍵 KDC_private_key_proprietary. をセカンダリ レルムのディレクトリにコピーします。たとえば、*BPR_HOME/kdc/solaris/packetcable/certificates* の下の */CISCO1.COM* ディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

m. セカンダリ CISCO2.COM レルムの KDC 証明書とそれに関連付けられる秘密鍵を作成します。

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

ツールについては、P.14-3 の「PKCert.sh ツールの使用方法」を参照してください。

n. *KDC.cer* をセカンダリ レルムのディレクトリにコピーします。たとえば、*BPR_HOME/kdc/solaris/packetcable/certificates* の下の */CISCO2.COM* ディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

o. 秘密鍵 KDC_private_key.pkcs8 をセカンダリ レルムのディレクトリにコピーします。たとえば、*BPR_HOME/kdc/solaris/packetcable/certificates* の下の */CISCO2.COM* ディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンドを参照してください。

p. 秘密鍵 KDC_private_key_proprietary. をセカンダリ レルムのディレクトリにコピーします。たとえば、*BPR_HOME/kdc/solaris/packetcable/certificates* の下の */CISCO2.COM* ディレクトリにコピーします。ファイルをコピーする方法については、Solaris のマニュアルの **cp** コマンド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 キーの生成に使用するパスワードは、**packetcable registration kdc service-key** コマンドを使用して DPE に設定したパスワードと一致するようにします。

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、P.14-9 の「KeyGen ツールの使用方法」を参照してください。

ステップ 7　ステップ 6 で生成したサービス キーは *BPR_HOME/kdc/solaris/keys* ディレクトリに保存してください。

次に例を示します。

```
/opt/CSCObac/kdc/solaris/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、Solaris のマニュアルを参照してください。

ステップ 8　各種証明書とサービス キーが *BPR_HOME/kdc* ディレクトリにあ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  solaris 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/solaris/logs
# ls
kdc.log    kdc.log.1

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

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

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

詳細については、Solaris の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、P.9-2 の「コマンドラインからの BAC プロセス ウォッチドッグの使用方法」を参照してください。

ステップ 10　複数レルム用の BAC の管理者のユーザ インターフェイスを設定します。

a.　セカンダリ レルム（この場合は CISCO1.COM）の DHCP 基準を追加します。

次に例を示します。

1. **Configuration** > **DHCP Criteria** > **Manage DHCP Criteria** で、**Add** ボタンをクリックします。
2. Add DHCP Criteria ページが表示されます。
3. DHCP Name フィールドに **cisco1** と入力します。
4. **Submit** をクリックします。
5. Manage DHCP Criteria ページに戻り、cisco1 DHCP criteria をクリックします。Modify DHCP Criteria ページが表示されます。
6. Property Name で、*/ccc/kerb/realm* を選択して、Property Value フィールドに CISCO1.COM と入力します。
7. **Add**、**Submit** の順にクリックします。

詳細については、P.13-15 の「DHCP 基準の設定」を参照してください。

b. セカンダリ レルム（この場合は CISCO2.COM）の DHCP 基準を追加します。

次に例を示します。

1. **Configuration** > **DHCP Criteria** > **Manage DHCP Criteria** で、**Add** ボタンをクリックします。
2. Add DHCP Criteria ページが表示されます。
3. DHCP Name フィールドに **cisco2** と入力します。
4. **Submit** をクリックします。
5. Manage DHCP Criteria ページに戻り、cisco2 DHCP criteria をクリックします。Modify DHCP Criteria ページが表示されます。
6. Property Name で、*/ccc/kerb/realm* を選択して、Property Value フィールドに cisco2.COM と入力します。
7. **Add**、**Submit** の順にクリックします。

詳細については、P.13-15 の「DHCP 基準の設定」を参照してください。

c. プロビジョニングする各デバイスの BAC にファイルとしてテンプレートを追加します。このステップでは、Motorola MTA 用に追加します。

次に例を示します。

1. **Configuration** > **Files** の順に選択します。Manage Files ページが表示されます
2. **Add** をクリックします。Add Files ページが表示されます。
3. File Type ドロップダウン リストから、CableLabs Configuration Template オプションを選択します。
4. *mot-mta.tmpl* 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するテンプレートです。テンプレートの構文については、例 7-2 を参照してください。
5. **Submit** をクリックします。

詳細については、P.13-18 の「ファイルの管理」を参照してください。

d. プロビジョニングする各デバイスの BAC にファイルとしてテンプレートを追加します。このステップでは、Linksys MTA 用に追加します。

次に例を示します。

1. **Configuration** > **Files** の順に選択します。Manage Files ページが表示されます
2. **Add** をクリックします。Add Files ページが表示されます。
3. File Type ドロップダウン リストから、CableLabs Configuration Template オプションを選択します。
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するテンプレートです。テンプレートの構文については、例 7-3 を参照してください。

5. **Submit** をクリックします。

詳細については、P.13-18 の「ファイルの管理」を参照してください。

e. プロビジョニングする各デバイスの BAC にファイルとしてテンプレートを追加します。このステップでは、SA MTA 用に追加します。

次に例を示します。

1. **Configuration** > **Files** の順に選択します。Manage Files ページが表示されます
2. **Add** をクリックします。Add Files ページが表示されます。
3. File Type ドロップダウン リストから、CableLabs Configuration Template オプションを選択します。
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するテンプレートです。テンプレートの構文については、例 7-4 を参照してください。
5. **Submit** をクリックします。

詳細については、P.13-18 の「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（この場合は CISCO.COM）のサービス クラスを追加します。

次に例を示します。

1. **Configuration** > **Class of Service** の順に選択します。
2. **Add** をクリックします。Add Class of Service ページが表示されます。
3. CISCO.COM レルムの新しいサービス クラスの名前として mot-mta と入力します。
4. サービス クラス タイプとして PacketCableMTA を選択します。
5. Property Name ドロップダウン リストから */cos/packetCableMTA/file* を選択して、*mot-mta.tmpl* テンプレート ファイル（プライマリ CISCO.COM レルムの Motorola MTA のプロビジョニングに使用）に関連付けます。
6. **Add**、**Submit** の順にクリックします。

詳細については、P.13-2 の「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（この場合は CISCO1.COM）のサービス クラスを追加します。

次に例を示します。

1. **Configuration** > **Class of Service** の順に選択します。
2. **Add** をクリックします。Add Class of Service ページが表示されます。
3. CISCO1.COM レルムの新しいサービス クラスの名前として linksys-mta と入力します。
4. サービス クラス タイプとして PacketCableMTA を選択します。
5. Property Name ドロップダウン リストから */cos/packetCableMTA/file* を選択して、*linksys-mta.tmpl* テンプレート ファイル（セカンダリ CISCO1.COM レルムの Linksys MTA のプロビジョニングに使用）に関連付けます。
6. **Add**、**Submit** の順にクリックします。

詳細については、P.13-2 の「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（この場合は CISCO2.COM）のサービス クラスを追加します。

次に例を示します。

1. **Configuration** > **Class of Service** の順に選択します。
2. **Add** をクリックします。Add Class of Service ページが表示されます。
3. CISCO1.COM レルムの新しいサービス クラスの名前として sa-mta と入力します。
4. サービス クラス タイプとして PacketCableMTA を選択します。
5. Property Name ドロップダウン リストから */cos/packetCableMTA/file* を選択して、*sa-mta.tmpl* テンプレート ファイル（セカンダリ CISCO2.COM レルムの SA MTA のプロビジョニングに使用）に関連付けます。
6. **Add**、**Submit** の順にクリックします。

詳細については、P.13-2 の「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にして、プロビジョニングします。プロビジョニング プロセスを説明してい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を説明します。

a. デバイスのケーブル モデム部分を、**sample-bronze-docsis** サービス クラスを使用するように設定して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場合は、**Devices** > **Manage Devices** ページに進みます。プロビジョニングする PacketCable デバイスを検索して選択します。Modify Device ページが表示されます。

c. ドメイン名を設定します。この例では bacclab.cisco.com を使用します。

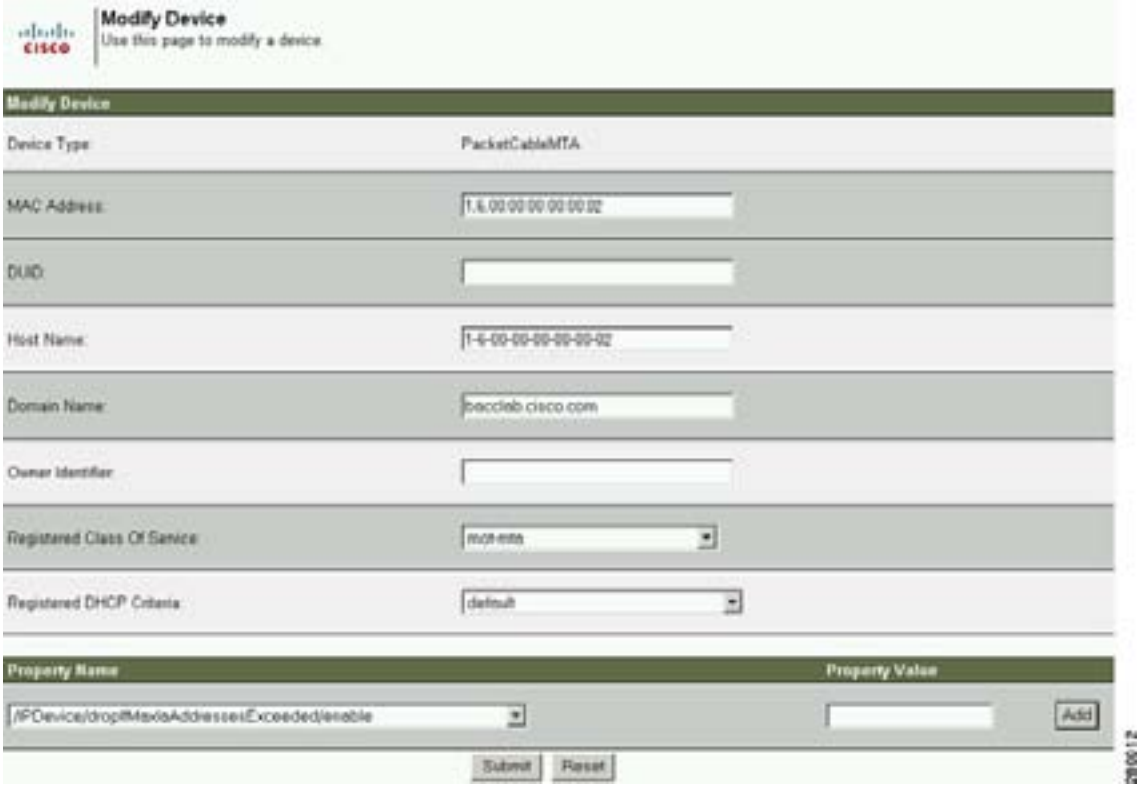
d. 登録されているサービス クラスに対応するドロップダウン リストから、**mot-mta** を選択します。これはステップ 10-f で追加したサービス クラスです。

e. 登録されている DHCP 基準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。

図 7-2 に、Motorola MTA のデバイスの詳細を示します。

図 7-2　Motorola MTA のプロビジョニング：デバイスの詳細



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. デバイスのケーブル モデム部分を、**sample-bronze-docsis** サービス クラスを使用するように設定して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場合は、**Devices > Manage Devices** ページに進みます。プロビジョニングする PacketCable デバイスを検索して選択します。Modify Device ページが表示されます。

c. ドメイン名を設定します。この例では bacclab.cisco.com を使用します。

d. 登録されているサービス クラスに対応するドロップダウン リストから、**linksys-mta** を選択します。これはステップ 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 基準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、ステップ 10-a でセカンダリ CISCO1.COM レルム用に追加した DHCP 基準です。

f. **Submit** をクリックします。

図 7-3 に、Linksys MTA のデバイスの詳細を示します。

図 7-3　Linksys MTA のプロビジョニング：デバイスの詳細

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. デバイスのケーブル モデム部分を、**sample-bronze-docsis** サービス クラスを使用するように設定して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場合は、**Devices > Manage Devices** ページに進みます。プロビジョニングする PacketCable デバイスを検索して選択します。Modify Device ページが表示されます。

c. ドメイン名を設定します。この例では bacclab.cisco.com を使用します。

d. 登録されているサービス 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。これはステップ 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 基準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、ステップ 10-b でセカンダリ CISCO2.COM レルム用に追加した DHCP 基準です。

f. **Submit** をクリックします。

図 7-4 に、SA MTA のデバイスの詳細を示します。

図 7-4　SA MTA のプロビジョニング：デバイスの詳細

ステップ 12 Ethereal トレースを使用して、複数レルムのサポートが動作可能かどうかを確認します。この手順で使用した設定例から表示された、KDC と DPE のログ ファイルの出力例を参照してください。

例 1

次の例は、プライマリ CISCO.COM レルムでプロビジョニングした Motorola SBV 5120 MTA の KDC および DPE ログ ファイルからの抜粋です。

KDC ログの出力例：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 ログの出力例：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

次の例は、セカンダリ CISCO1.COM レルムでプロビジョニングした Linksys CM2P2 MTA の KDC および DPE ログ ファイルからの抜粋です。

KDC ログの出力例：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 ログの出力例：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

次の例は、セカンダリ CISCO2.COM レルムでプロビジョニングした SA WebStar DPX 2203 MTA の KDC および DPE ログ ファイルからの抜粋です。

KDC ログの出力例：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 ログの出力例：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]
```

複数レルムのデバイスのプロビジョニングに使用するテンプレートのオーサリング

ここで説明するテンプレートの構文を使用して、特定レルムのデバイスをプロビジョニングできます。ここで示す例は、Motorola SBV5120 MTA（例 7-2）、Linksys CM2P2 MTA（例 7-3）、および SA WebStar DPX2203 MTA（例 7-4）に固有のものです。

（注）これらのテンプレートを修正して、自分のネットワーク内の MTA の仕様に合せる必要があります。

例 7-2　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,ST
RING,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,S
TRING,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,"
'43:61:62:6C:65:4C:61:62:73:2C:20:49:6E:63:2E'"
#
# 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',STRIN
G,CISCO.COM
#
# Finally, the end marker.
#
option 254 255
```

例 7-3　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 に設定されている点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,ST
RING,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,S
TRING,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,
"'43:61:62:6C:65:4C:61:62:73:2C:20:49:6E:63:2E'"
#
# 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',STRIN
G,CISCO1.COM
#
# Finally, the end marker.
#
option 254 255
```

例 7-4　SA 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 CISCO2.COM に設定されている点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,ST
RING,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,S
TRING,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,
"'43:61:62:6C:65:4C:61:62:73:2C:20:49:6E:63:2E'"
#
# 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',STRIN
G,CISCO2.COM
#
# Finally, the end marker.
#
option 254 255
```

Network Registrar DNS サーバでの SRV レコードの設定

Network Registrar DNS サーバが KDC とともに稼動するように設定する必要があります。この設定を行う場合は、Network Registrar のマニュアルとその説明を参照してください。

（注） 目的のレルム名と一致するゾーン名を作成し、この特殊なゾーン内の DNS レコード（ゾーンを維持するために DNS サーバが必要とするレコード以外）のみをそのレルムの SRV レコードにすることをお勧めします。この例では、目的の Kerberos レルムが voice.example.com で、その他のすべての KDC、Network Registrar、および DPE の設定がすでに行われていることを想定しています。KDC の FQDN は、kdc.example.com であると仮定します。

ステップ 1　**nrcmd** コマンドライン ツールを起動します（デフォルトでは、*/opt/nwreg2/local/usrbin* ディレクトリにあります）。

ステップ 2　ユーザ名とパスワードを入力します。

ステップ 3　Kerberos レルムのゾーンを作成するには、次のように入力します。

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

ここで、*address_of_nameserver* はネーム サーバの IP アドレスを指定します。

ステップ 4　SRV レコードを新しいゾーンに追加するには、次のように入力します。

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

ここで、*KDC_FQDN* は KDC の FQDN を指定します。

ステップ 5　保存して、DNS サーバをリロードするには、次のように入力します。

```
nrcmd> save
nrcmd> dns reload
```

PacketCable MTA と安全に通信するための RDU と DPE 上での SNMPv3 クローニングの設定

BAC を使用すると、SNMPv3 の外部ネットワーク マネージャから MTA デバイスへのアクセスをイネーブルにできます。また、RDU は特定の MTA で SNMPv3 処理を実行できます。

この機能をイネーブルにするには、DPE と RDU にセキュリティ鍵関連情報を設定します。鍵関連情報が設定されると、クローニングされた SNMPv3 エントリの作成に使用する BAC Application Programming Interface（API; アプリケーション プログラミング インターフェイス）コールがイネーブルになります。

（注） この機能をイネーブルにすると、プロビジョニングのパフォーマンスに影響を与えます。

鍵関連情報の作成と鍵の生成

鍵関連情報を作成するには、次の 2 つのステップの処理を実行します。

1. RDU でスクリプト コマンドを実行します。
2. DPE で CLI コマンドを実行します。

（注） この共有秘密情報は、CMTS または BAC の共有秘密情報と同じものではありません。

鍵関連情報を作成するには、次の手順に従います。

ステップ 1　*BPR_HOME/rdu/bin* ディレクトリから、次のスクリプトを RDU 上で実行します。

```
# generateSharedSecret.sh password
```

password は、ユーザが作成する 6 ～ 20 文字の任意のパスワードです。このパスワードは、46 バイトの鍵の生成に使用されます。この鍵は、*keymaterial.txt* という名前のファイルに保存されます。このファイルは *BPR_HOME/rdu/conf* ディレクトリにあります。

ステップ 2　ステップ 1 でこの鍵の生成に使用した *password* を使用して、この音声技術がイネーブルになっているすべての DPE 上で **service packetcable** *1..1* **snmp key-material** DPE CLI コマンドを実行します。このコマンドは、DPE 上に同じ 46 バイトの鍵を生成し、RDU と DPE を同期して、MTA と安全に通信できるようにします。

PacketCable eMTA の Basic プロビジョニング

BAC は簡潔で DOCSIS に似た、ノンセキュアなプロビジョニング フローを提供する PacketCable Basic もサポートしています。表 7-5 で、図 7-1 のプロビジョニング ワークフローを使用する BASIC.1 のフローを説明します。

表 7-5　PacketCable eMTA の Basic プロビジョニング

手順	ワークフロー	説明
MTA-1	DHCP ブロードキャスト検出	セキュアなフローと同様に実行されます。
MTA-2	DHCP オファー	プロビジョニング システムが BASIC.1 モードで MTA をプロビジョニングするように設定されている場合、プロビジョニング システムは、Option 122 のサブオプション 6（特別な予約済みレルム名「BASIC.1」を含む）を含んでいる DHCP オファーを返します。この予約済みレルム名は、BASIC.1 プロビジョニング フローを使用するように MTA に指示します。このオファーには、Option 122.3 のプロビジョニング システムの IP アドレスも含まれており、file フィールドと siaddr フィールドには MTA 設定ファイルがある場所が入力されています。
MTA-3	DHCP 要求	MTA DHCP 交換の残りが実行されます（要求と確認の交換）。
MTA-4	DHCP 確認	
MTA-22	テレフォニー設定ファイル要求	MTA は、ステップ MTA-22 まで直接スキップします。file と siaddr の情報を使用して、MTA はその設定ファイルを TFTP 経由でプロビジョニング システムからコピーします。BAC は、TFTP サーバを DPE コンポーネントに統合します。 **（注）** MTA、プロビジョニング サーバの認証、または暗号化は一切発生しません。
MTA-23	テレフォニー設定ファイル	

BASIC.2 フローは、次の例外を除いて BASIC.1 と同一です。

- 「BASIC.2」が MTA の DHCP Option 122 のサブオプション 6 に入力される。
- フローの最後の MTA-25 で、MTA がプロビジョニング ステータス SNMPv2c INFORM を発行する（DHCP Option 122 のサブオプション 3 が通知対象を指定）。

PacketCable Basic フローは DOCSIS フローと似ていますが、次の点が異なります。

- MTA とプロビジョニング システム間で ToD 交換が行われない。
- MTA 設定ファイルに完全性ハッシュが含まれる。具体的に、設定ファイルの内容全体の SHA1 ハッシュが pktcMtadevConfigFileHash SNMP VarBind に入力され、EoF TLV ファイル直前の TLV 11 に配置される。
- MTA がその設定ファイルを受信して処理すると、BASIC.2 フローがプロビジョニング ステータス SNMPv2c 通知を発行する。この通知は、MTA のプロビジョニングが正常に完了したかどうかを BAC に通知します。問題が発生した場合は、エラーが生成され、イベントが DPE から RDU に、そして BAC クライアントに送信されます。この通知は、設定ファイルの問題をデバッグする場合に役立ちます。

DOCSIS フローの詳細については、第 6 章「DOCSIS 設定」を参照してください。

（注）PacketCable Basic のプロビジョニング フローを使用する前に、PacketCable Basic に対応した eMTA を使用していることを確認してください。eMTA は、DHCP 検出の Option 60、TLV 5.18（サポートされているフロー）で Basic 対応であることを報告する必要があります。

PacketCable TLV 38 および MIB のサポート

BAC は一連の PacketCable 1.5 MIB を完全にサポートしています。

BAC は PacketCable 設定テンプレートで TLV 38 をサポートしています。この TLV を使用して、複数の SNMP 通知ターゲットを設定できます。この TLV を設定すると、TLV 38 で設定したターゲットにもすべての通知が発行されることになります。

SNMP v2C の通知

BAC は PacketCable MTA からの SNMP v2C TRAP と INFORM 通知の両方をサポートします。

Euro PacketCable

Euro-PacketCable サービスは、基本的に北米版 PacketCable サービスの欧州版に相当しますが、次の点が異なります。

- Euro PacketCable では、使用する MIB が異なる。
- Euro PacketCable では、使用するデバイス証明書（*MTA_Root.cer*）とサービス プロバイダー証明書（サービス プロバイダーのルート）のセットが異なる。

  Euro-PacketCable 証明書の場合、*kdc.ini* ファイルの *euro-packetcable* プロパティを true に設定する必要があります。KDC は、Euro-PacketCable（tComLabs）証明書チェーンをサポートしています。次は、Euro PacketCable がイネーブルになっている KDC 設定ファイルの例です。

  ```
  [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
  ```

  Euro PacketCable を使用する場合は、PacketCable のプロパティ */pktcbl/prov/locale* の値を必ず EURO に設定してください。デフォルトは NA（North America）です。ロケールは、設定ファイル ユーティリティで指定できます。詳細については、P.5-23 の「設定ファイル ユーティリティの使用方法」を参照してください。

Euro-PacketCable MIB

Euro-PacketCable MIB は、基本的にはドラフト IETF MIB のスナップショットです。MTA 設定ファイルは、MIB を参照する SNMP VarBinds で構成されます。北米版 PacketCable と Euro-PacketCable の MIB は大きく異なるため、北米版 PacketCable と Euro-PacketCable の設定ファイルには互換性はありません。インストール時に、北米版 PacketCable のサンプル ファイル（*cw29_config.tmpl*）と Euro PacketCable のサンプル ファイル（*ecw15_mta_config.tmpl*）が *BPR_HOME/rdu/samples* ディレクトリにコピーされます。

BAC には次の Euro-PacketCable MIB が付属しています。

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

Euro-PacketCable MIB の設定

Euro-PacketCable MIB を使用するように BAC を設定するには、ロードする MIB を指定する BAC RDU プロパティを変更する必要があります。デフォルトでは、このプロパティには PacketCable MIB が含まれます。

次のいずれかの方法でプロパティを変更できます。

- *rdu.properties* を修正して、RDU を再起動する。
- 管理者のユーザ インターフェイスで、**Configuration** > **Defaults** > **System Defaults** に移動して、MIB リストを下に示すリストで置き換えます。RDU を再起動する必要はありません。
- Prov API *changeSystemDefaults()* コールを使用する。RDU を再起動する必要はありません。

プロパティ名は */snmp/mibs/mibList*（properties ファイル）または `SNMPPropertyKeys.MIB_LIST`（Prov API 定数名）です。プロパティの値は、Comma-Separated Value（CSV; カンマ区切り形式）で、次に示す必須の MIB 名で構成されます。

```
/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-CABL
E-SPECTRUM-MIB,CISCO-DOCS-EXT-MIB,SNMP-FRAMEWORK-MIB,DOCS-CABLE-DEVICE-MIB,DOCS-CABLE-
DEVICE-MIB-OBSOLETE,DOCS-QOS-MIB,CISCO-CABLE-MODEM-MIB,DOCS-IETF-BPI2-MIB,INTEGRATED-S
ERVICES-MIB,DIFFSERV-DSCP-TC,DIFFSERV-MIB,TCOMLABS-MIB,PKTC-TCOMLABS-MTA-MIB,PKTC-TCOM
LABS-SIG-MIB
```

CHAPTER 8

CableHome の設定

この章では、適切な CableHome 配備を実現するために必要となる作業について説明します。CableHome テクノロジーには、セキュア（SNMP）とノンセキュア（DHCP）の 2 つのバージョンがあります。この章では、主にノンセキュア バージョンについて取り上げます。

この章は、ユーザが CableHome の仕様（CH-SP-CH1.0-I05-030801）の内容に精通していることを想定しています。

ノンセキュア CableHome プロビジョニングのフロー

ノンセキュア CableHome プロビジョニング フローでどのステップが失敗したかを特定してから、他の詳細事項を診断するようにすると、非常に便利です。図 8-1 に、主要なプロビジョニング フローの要約を示します。

図 8-1　ノンセキュア CableHome のフロー

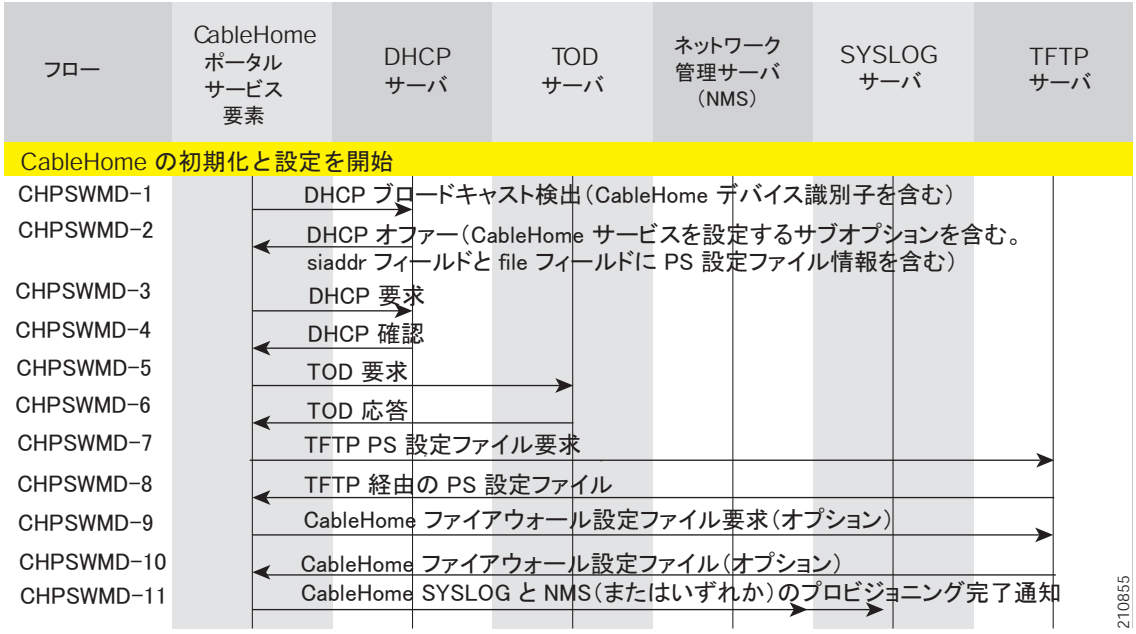


表 8-1 で、ノンセキュアな CableHome の配備のプロビジョニング フローについて説明します。

表 8-1　CableHome プロビジョニング ワークフロー

手順	ワークフロー	説明
CHPSWMD-1	DHCP 検出	WAN-MAN が、自身の IP リースを取得します。
CHPSWMD-2	DHCP オファー	プロビジョニング システムが、次の CableHome Option 177 サブオプションとともに DHCP オファーを返します。 • 3：サービス プロバイダーの SNMP Entity Address を指定します。 • 6：プロビジョニング レルムの Kerberos レルム名を指定します。レルム名は、ポータル サービスが Key Distribution Center（KDC; 鍵発行局）のアドレスに対する DNS ルックアップを許可するために必要となります。 • 51：Kerberos サーバの IP アドレスを指定します。このサーバは、1 つ以上の Key Distribution Center サーバのネットワーク アドレスのポータル サービスを通知します。 このオファーの file フィールドと siaddr フィールドには、ポータル サービスを設定するのに必要なファイル情報も含まれます。
CHPSWMD-3	DHCP 要求	ポータル サービスが、DHCP オファーを受け入れるように、適切な DHCP サーバに DHCP 要求メッセージを送信します。
CHPSWMD-4	DHCP 確認	DHCP サーバが、ポータル サービスの IPv4 アドレスを含む DHCP 確認を返します。DHCP 確認で受信した情報に基づき、ポータル サービスが DHCP（ノンセキュア）モードでのプロビジョニングを指定する、`cabhPsDevProvMode` パラメータを修正します。また、Time of Day サーバのアドレスが `cabhPsDevTimeServerAddr` パラメータに保存されます。
CHPSWMD-5	ToD 要求	ポータル サービスが、DHCP 確認メッセージの Option 4 で指定されているタイム サーバとの Time of Day 同期を開始します。
CHPSWMD-6	ToD 応答	Time of Day サーバが現在時刻を UTC 形式で応答します。
CHPSWMD-7	TFTP による PS 設定ファイル	ポータル サービスが、設定ファイルを取得するために TFTP 取得要求を送信します。
CHPSWMD-8	CableHome ファイアウォール設定ファイル要求	設定ファイルが TFTP 経由でダウンロードされます。オプションで、ロードするファイアウォール設定があり、その設定を指定するためにこの方式が選択されている場合、IP アドレスの名前とファイアウォール設定ファイルのハッシュが設定ファイルに含まれます。

表 8-1　CableHome プロビジョニング ワークフロー（続き）

手順	ワークフロー	説明
CHPSWMD-9	CableHome ファイアウォール設定ファイル要求	ステップ CHPSWMD-8 で取得した設定ファイルにファイアウォール情報が含まれている場合、ポータル サービスは、ファイアウォール設定 TFTP サーバへの TFTP 取得要求経由のファイアウォール設定ファイルを取得する場合もあります。 設定ファイルにファイアウォール設定情報がない場合は、プロビジョニング プロセスはステップ CHPSWMD-9 と CHPSMWD-10 をスキップします。
CHPSWMD-10	CableHome ファイアウォール設定ファイル	ファイアウォール設定 TFTP サーバが、ファイアウォール設定ファイルを含む TFTP 応答を送信します。
CHPSWMD-11	CableHome SYSLOG またはプロビジョニング完了の NMS 通知（または両方）	設定が正常に終了すると、ポータル サービスは、正常に設定されたことを BAC に通知するために、syslog メッセージ、SNMP トラップ、または両方を送信します。

CableHome の設定

この項では、Cisco Network Registrar、および Cable Modem Termination System（CMTS; ケーブル モデム ターミネーション システム）の設定方法について説明します。

Network Registrar の設定

ステップ 1　プロビジョニングされる WAN-MAN とプロビジョニングされない WAN-MAN、さらにプロビジョニングされる WAN-Data の選択タグも作成します。

ケーブル モデムについて、プロビジョニングされないクライアント クラスとプロビジョニングされるクライアント クラス、およびスコープを『*User Guide for Cisco Network Registrar 7.0*』で指定されているように設定します。

ステップ 2　WAN-MAN について、プロビジョニングされないクライアント クラスとプロビジョニングされるクライアント クラス、およびスコープを設定します。

ステップ 3　WAN-Data について、プロビジョニングされるクライアント クラスとスコープを設定します。

ステップ 4　すべてのサブネットに到達するためのルートを追加します。

RDU の設定

RDU に対する CableHome サポートを設定するには、次の設定を行います。

- CableHome WAN-MAN の設定（P.8-4）
- CableHome WAN-Data の設定（P.8-4）

CableHome WAN-MAN の設定

1. プロビジョニングされる WAN-MAN の DHCP 基準を作成します。作成するには、クライアント クラスを Network Registrar CableHome WAN-MAN で設定されているクライアント クラス名に設定します。
2. プロビジョニングされる WAN-MAN のサービス クラスを作成します。
   - */cos/chWanMan/file* を、このサービス クラス用の適切な CableHome 設定ファイルに設定します。
   - */chWanMan/firewall/file* を、目的のファイアウォール設定ファイルに設定します。

CableHome WAN-Data の設定

ポータル サービスで WAN-Data の IP アドレスを取得するときは、そのたびに次の WAN-Data パラメータを設定します。

1. WAN-Data の DHCP 基準を作成します。
2. WAN-Data のサービス クラスを作成します。

DPE の設定

CableHome テクノロジーをサポートするように DPE を設定するには、次の手順に従います。

ステップ 1　CableHome デバイス プロビジョニング WAN-MAN 設定ファイルを開いて、DHCP Option 60 が CableHome1.0 または CableHome1.1 のいずれかに設定されていることを確認します。一部の製造業者では、独自の MIB オブジェクトを使用して、単純なケーブル モデム、CableHome 以外のルータ、または CableHome ルータとして動作するようにデバイスに指示していることがあります。デバイスの DHCP パケットの DHCP Option 60 に CableHome1.0 または CableHome1.1 が含まれていない場合、デバイスは必ずコンピュータとして表示されます。

ステップ 2　ポータル サービスで WAN-Data の IP アドレスを取得する場合は、次の手順に従います。

- WAN-MAN 設定ファイルに含まれる TLV 28 で、cabhCdpWanDataIpAddrCount が 0 を超える値に設定されていることを確認します。
- ケーブル モデム設定ファイルで、デバイスの最大数を WAN-Data IP アドレス数を収容できる値に設定します。

ステップ 3　CableHome デバイスのブート時にセルフプロビジョニングをイネーブルにするには、次の手順に従います。

- *unprov-wan-man.cfg* ポータル サービス設定ファイルで、ポータル サービスをパススルー モードで設定します。
- ケーブル モデム設定ファイルで、デバイスの最大数を 2 以上に設定して、WAN-MAN とコンピュータをプロビジョニングできるようにします。このコンピュータは、サインアップ Web ページに直接アクセスしてセルフプロビジョニングできます。

CHAPTER 9

Broadband Access Center の管理

この章では、Broadband Access Center（BAC）システムの管理に役立つ各種サブコンポーネントについて説明します。この章は、次の項で構成されています。

BAC プロセス ウォッチドッグ

BAC プロセス ウォッチドッグは、すべての BAC プロセスのランタイム状況を監視する管理エージェントです。このプロセス ウォッチドッグにより、プロセスが予想外に停止した場合に自動的に再開されるようになります。BAC コンポーネントを実行する各システム上で、BAC プロセス ウォッチドッグのインスタンスが 1 つ実行されます。

BAC プロセス ウォッチドッグは、監視対象プロセスの状態を開始、停止、再開、決定するコマンドライン ツールとして利用できます。

監視対象のアプリケーションが機能しなくなると、自動的に再開されます。何らかの理由で再開プロセスも機能しない場合は、BAC プロセス ウォッチドッグ サーバは所定の時間待機してから再開を試みます。

（注） Cisco Network Registrar にインストールされている拡張を監視するために、BAC プロセス ウォッチドッグおよび SNMP エージェントを使用する必要はありません。

再開を試みる間隔は 1 秒から始まり、後続の試行で 5 分に達するまで指数関数的に長くなります。その後、プロセスの再開が成功するまで 5 分間隔で試みられます。再開の成功の 5 分後に、期間は再び自動的に 1 秒にリセットされます。

次に例を示します。

1. プロセス A が失敗します。
2. BAC プロセス ウォッチドッグ サーバはプロセスの再開を試み、1 回目の再開が失敗します。
3. BAC プロセス ウォッチドッグ サーバは 2 秒間待機してからプロセスの再開を試み、2 回目の再開が失敗します。
4. BAC プロセス ウォッチドッグ サーバは 4 秒間待機してからプロセスの再開を試み、3 回目の再開が失敗します。
5. BAC プロセス ウォッチドッグ サーバは 16 秒間待機してからプロセスの再開を試みます。

コマンドラインからの BAC プロセス ウォッチドッグの使用方法

BAC プロセス ウォッチドッグは、システムのブートアップのたびに自動的に起動します。そのため、このウォッチドッグは、同じシステムにインストールされている BAC システム コンポーネントも起動します。**/etc/init.d/bprAgent** コマンドを実行すると、単純なコマンドライン ユーティリティを使用して BAC ウォッチドッグを制御することができます。

表 9-1 は、BAC プロセス ウォッチドッグに対して使用できる Command-Line Interface（CLI; コマンドライン インターフェイス）コマンドを示しています。

表 9-1　BAC CLI コマンド

コマンド	説明
bprAgent start	すべての監視対象プロセスを含む BAC プロセス ウォッチドッグを開始します。
bprAgent stop	すべての監視対象プロセスを含む BAC プロセス ウォッチドッグを中止します。
bprAgent restart	すべての監視対象プロセスを含む BAC プロセス ウォッチドッグを再起動します。

表 9-1　BAC CLI コマンド（続き）

コマンド	説明
bprAgent status	すべての監視対象プロセスを含む BAC プロセス ウォッチドッグの状態を入手します。
bprAgent start *process-name*	特定の 1 つの監視対象プロセスを開始します。*process-name* 値がそのプロセスを識別します。
bprAgent stop *process-name*	特定の 1 つの監視対象プロセスを中止します。*process-name* 値がそのプロセスを識別します。
bprAgent restart *process-name*	特定の 1 つの監視対象プロセスを再開します。*process-name* 値がそのプロセスを識別します。
bprAgent status *process-name*	特定の 1 つの監視対象プロセスの状態を入手します。*process-name* 値がそのプロセスを識別します。

この表に示す *process-name* は、次のいずれかになります。

- **rdu**：RDU サーバを指定します。
- **dpe**：DPE サーバを指定します。
- **kdc**：KDC サーバを指定します。
- **snmpAgent**：SNMP エージェントを指定します。
- **tomcat**：管理者を指定します。
- **cli**：DPE CLI を指定します。

（注） Solaris オペレーティング システムがリブートされると、BAC プロセス ウォッチドッグが最初に停止します。その結果、BAC サーバは正常にシャットダウンできます。オペレーティング システムをシャットダウンまたはリブートするには、Solaris **shutdown** コマンドを使用してください。Solaris **reboot** コマンドでは、アプリケーション シャットダウン フックは実行されません。BAC プロセスはシャットダウンされるのではなく、強制終了されるので注意してください。この処理は BAC に悪影響を与えるものではありませんが、場合によっては、サーバの起動が遅くなったり、特定の統計情報やパフォーマンス カウンタに歪みが生じたりすることがあります。

BAC ウォッチドッグ デーモンでアクションをトリガーするイベント（プロセス障害および再起動を含む）は、ログ ファイル *BPR_HOME/agent/logs/agent.log* に記録されます。ウォッチドッグ デーモンでは、重要なイベントも標準の `local6` ファシリティの下の syslog に記録されます。

管理者のユーザ インターフェイス

BAC 管理者のユーザ インターフェイスは、BAC システムを集中管理するための Web ベースのアプリケーションです。このシステムを使用して、次の作業を行うことができます。

- グローバル デフォルトの設定
- カスタム プロパティの定義
- サービス クラスの追加、修正、および削除
- DHCP 基準の追加、修正、および削除
- デバイスの追加、修正、および削除
- デバイス情報の追加および編集
- デバイスのグループ化
- サーバの状態とサーバ ログの表示
- ユーザの管理

このインターフェイスの使用方法については、それぞれ次の章を参照してください。

- 第 11 章「管理者のユーザ インターフェイスについて」：BAC 管理者のユーザ インターフェイスにアクセスする方法や設定方法について説明します。
- 第 12 章「管理者のユーザ インターフェイスの使用方法」：各種 BAC コンポーネントの監視など、管理作業を行う方法について説明します。
- 第 13 章「Broadband Access Center の設定」：BAC を設定するために実行する作業について説明します。

コマンドライン インターフェイス

BAC CLI は、Telnet または SSH を使用して DPE を設定したり、DPE の状態を表示したりするために使用する、IOS に似たコマンドライン インターフェイスです。CLI では、組み込み型のコマンド ヘルプとコマンドのオートコンプリート機能がサポートされています。

CLI の認証は、ローカルで設定したログイン パスワードと特権パスワード、または TACACS+ サービスのリモート ユーザ名とパスワードを使用してイネーブルにできます。

DPE CLI にアクセスするには、ローカル ホストまたはリモート ホストからポート 2323 への Telnet セッションを開きます。

ローカル ホストから DPE CLI へのアクセス

ローカル ホストから CLI にアクセスするには、次のコマンドを使用できます。

telnet *local_hostname* ***2323***

または

telnet 0 2323

リモート ホストから DPE CLI へのアクセス

リモート ホストから CLI にアクセスするには、次のコマンドを入力します。

telnet *remote-hostname* ***2323***

（注） CLI への Telnet 接続を確立できない場合は、CLI サーバが稼働していない可能性があります。その場合は、次のように入力してサーバを起動します。
 # /etc/init.d/bprAgent start cli

CLI にアクセスした後、続行するには DPE パスワードを入力する必要があります。デフォルトのログイン パスワードと特権パスワードは **changeme** です。

DPE がサポートする CLI コマンドの詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

SNMP エージェント

BAC では、RDU サーバおよび DPE サーバについて基本的な SNMP v2 ベースの監視がサポートされます。BAC SNMP エージェントでは SNMP 通知と SNMP トラップがサポートされます。それらをまとめて「通知」と呼びます。snmp-server CLI コマンドを使用して DPE に SNMP エージェントを設定し、SNMP 設定コマンドライン ツールを使用して RDU に SNMP エージェントを設定できます。

SNMP 設定コマンドライン ツールの詳細については、P.10-11 の「snmpAgentCfgUtil.sh ツールの使用方法」を参照してください。DPE CLI の詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

BAC ツール

BAC には、特定の機能をより効率的に実行するための自動ツールが用意されています。

表 9-2 は、この BAC リリースでサポートされている各種ツールを示しています。

表 9-2　BAC ツール

ツール	説明	参照先
設定ファイル ユーティリティ	BAC のテンプレートと設定ファイルをテスト、検証、および表示するために使用されます。	設定ファイル ユーティリティの使用方法（P.5-23）
BAC プロセス ウォッチドッグ	BAC ウォッチドッグ デーモンと連動して BAC システム コンポーネントの状態を監視し、サーバを停止または起動します。	コマンドラインからの BAC プロセス ウォッチドッグの使用方法（P.9-2）
RDU ログ レベル ツール	RDU のログ レベルを設定し、デバッグ　ログ出力をイネーブルまたはディセーブルにします。	RDU ログ レベル ツールの使用方法（P.10-5）
PacketCable 証明書ツール	機能するために KDC で必要とされる KDC 証明書をインストールおよび管理します。	PKCert.sh ツールの使用方法（P.14-3）
KeyGen ツール	PacketCable サービス キーを生成します。	KeyGen ツールの使用方法（P.14-9）
Network Registrar のプロパティ変更ツール	Network Registrar DHCP サーバに組み込まれている BAC 拡張で使用される主要な設定プロパティを変更するために使用します。	changeNRProperties.sh ツールの使用方法（P.14-11）
SNMP エージェント設定ツール	SNMP エージェントを管理します。	snmpAgentCfgUtil.sh ツールの使用方法（P.10-11）
診断ツール	システム パフォーマンスおよびトラブルシューティングに関連するサーバ データを収集します。	診断ツールによるトラブルシューティング（P.16-6）
BundleState.sh ツール	サポート拡大のサーバ状態に関連した診断データを組み込みます。	サポートを受けるためのサーバ状態のバンドル（P.16-11）
ディスク容量監視ツール	1 つまたは複数のファイル システムのしきい値を設定します。これらのしきい値を超えると、追加のディスク領域が利用可能になるまでアラートが生成されます。	disk_monitor.sh ツールの使用方法（P.14-13）

CHAPTER 10

Broadband Access Center の監視

この章では、Broadband Access Center（BAC）配備内の中央 RDU サーバおよび DPE サーバを監視する方法について説明します。次のトピックについて説明します。

イベントのロギング

イベントのロギングは RDU と DPE で実行されます。まれに、視認性向上のために、DPE イベントが RDU に記録されることもあります。

ログ ファイルはそれぞれのログ ディレクトリに保存され、任意のテキスト エディタを使用して調べることができます。ログ ファイルを圧縮すると、トラブルシューティングや障害の解決のために Cisco Technical Assistance Center またはシステム インテグレータに電子メールで送信しやすくなります。

また、RDU と DPE のログには、管理者のユーザ インターフェイスからアクセスすることもできます。

ログのレベルおよび構造

例 10-1 に示されているログ ファイル構造には、次のものが含まれます。

- Domain Name：ログ ファイルが生成されたコンピュータの名前。
- Date and Time：メッセージがログに記録された日時。ここには、該当するシステムの時間帯も示されます。
- Facility：システムを識別します（この場合は BAC）。
- Sub-facility：BAC のサブシステムまたはコンポーネントを識別します。
- Severity Level：ログ システムが定義する、ログの問題を処理するときの緊急性を識別するために使用される 7 段階の重大度のレベル（表 10-1 を参照）。次の重大度のレベルの設定方法については、P.10-3 の「重大度のレベルの設定」を参照してください。

表 10-1　重大度のレベル

ログ レベル	説明
0：緊急	システムが不安定です。すべての緊急メッセージを保存するように、ロギング機能を設定します。
1：アラート	すぐに対応が必要です。すぐに対応が必要なすべてのアクティビティ、およびさらに深刻な活動を保存するように、ロギング機能を設定します。
2：クリティカル	クリティカルな状態が存在します。すべてのエラー メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
3：エラー	エラー状態が存在します。すべてのエラー メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
4：警告	警告状態が存在します。すべての警告メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
5：通知	通常ですが、重大な状態が存在します。すべての通知メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
6：情報	情報メッセージ。利用可能なすべてのロギング メッセージを保存するように、ロギング機能を設定します。

（注） 7（デバッグ）として知られるもう 1 つのレベルは、シスコでデバッグの目的にのみ使用されます。Cisco Technical Assistance Center から指示された場合を除き、このレベルは使用しないようにしてください。

- Msg ID：メッセージ テキストの固有な識別子。
- Message：実際のログ メッセージ。

例 10-1　ログ ファイルのサンプル

Domain Name	Data and Time	Facility	Sub-facility	Severity Level	Msg ID	Message
bac.example.com:	2007 9 6 03:06:11 EST:	%BAC-	RDU-	5	0236:	Broadband Access Center Regional Distribution Unit starting up
bac.example.com:	2007 9 6 03:06:15 EST:	%BAC-	RDU-	5	0566:	Initialized API defaults
bac.example.com:	2007 9 6 03:06:15 EST:	%BAC-	RDU-	5	0567:	Initialized Network Registrar defaults
bac.example.com:	2007 9 6 03:06:15 EST:	%BAC-	RDU-	5	0568:	Initialized Server defaults
bac.example.com:	2007 9 6 03:06:18 EST:	%BAC-	RDU-	5	0570:	Initialized DOCSIS defaults
bac.example.com:	2007 9 6 03:06:18 EST:	%BAC-	RDU-	5	0571:	Initialized Computer defaults
bac.example.com:	2007 9 6 03:06:19 EST:	%BAC-	RDU-	5	0573:	Initialized CableHome WAN-MAN defaults
bac.example.com:	2007 9 6 03:06:19 EST:	%BAC-	RDU-	5	0572:	Initialized PacketCable defaults
bac.example.com:	2007 9 6 03:06:19 EST:	%BAC-	RDU-	5	0569:	Created default admin user
bac.example.com:	2007 9 6 03:06:20 EST:	%BAC-	RDU-	5	0575:	Database initialization completed in [471] msec
bac.example.com:	2007 9 6 03:06:25 EST:	%BAC-	RDU-	3	0015:	Unable to locate manifest file
bac.example.com:	2007 9 6 03:06:28 EST:	%BAC-	RDU-	3	0280:	Command error

重大度のレベルの設定

RDU と DPE のログの重大度のレベルは、いずれも特定の要件に合せて設定できます。たとえば、RDU の重大度のレベルを「警告」、DPE の重大度のレベルを「アラート」に設定できます。

ログ メッセージは、特定のイベントの発生に基づいて記述されます。イベントが発生するたびに、該当するログ メッセージと重大度のレベルが割り当てられます。重大度のレベルが設定したレベル以下であれば、メッセージがログに書き込まれます。レベルが設定した値より高い場合、メッセージはログに書き込まれません。

たとえば、ログ レベルが 4（警告）に設定されているとします。ログ ファイルには、ログ レベルが 4 以下に設定されているイベントの生成するメッセージがすべて書き込まれます。ログ レベルが 6（情報）に設定されている場合、ログ ファイルにはすべてのメッセージが書き込まれます。したがって、ログ レベルを高く設定するほど、ログ ファイルのサイズは大きくなります。

（注） KDC は、このログ ファイルでは考慮されません。

DPE に対する重大度のレベルを設定するには、DPE コマンドラインから **log level** コマンドを使用します。詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

RDU でのログ レベル ツールを設定するには、P.10-5 の「RDU ログ レベル ツールの使用方法」を参照してください。

ログ ファイルの循環

すべてのログ ファイルには、設定済みの最大ファイル サイズに基づいて番号が付けられ、ロールオーバーされます。デフォルトの最大ファイル サイズは 25 MB です（Application Programming Interface（API; アプリケーション プログラミング インターフェイス）から最大ファイル サイズを設定するには、*ServerDefaultsKeys.SERVER_LOG_MAXSIZE* プロパティを使用します）。ログ ファイルが設定済みの制限に達すると、データは別のファイルにロール オーバーされます。このファイルの名前は、*XXX.N*.log という形式で変更されます。内容は次のとおりです。

- *XXX*：ログ ファイルの名前を指定します。
- *N*：1 ～ 200 のいずれかの値を示します。

（注） RDU サーバと DPE サーバは、一時に最大 200 ログ ファイルを保存します。これらのサーバのログ ファイルのリストについては、後続の項を参照してください。

たとえば、*rdu.log* が 25 MB の制限に達すると、このファイルの名前は *rdu.1.log* に変更されます。ファイルのサイズが 25 MB 増えるたびに、最新のファイルの名前は *rdu.2.log*、*rdu.3.log* のように変更されます。したがって、*rdu.4.log* ファイルには、*rdu.7.log* より新しいデータが含まれます。ただし、最新のログ情報が保存されているのは、常に *rdu.log* です。

RDU のログ

RDU には次の 2 つのログがあり、*BPR_DATA/rdu/logs* ディレクトリで保持されます。

- *rdu.log*：設定されたデフォルトの重大度のレベルに従って、RDU 処理を記録します。（デフォルトのログ レベルの設定方法については、P.10-6 の「RDU ログ レベルの設定」を参照してください）。
- *audit.log*：BAC の設定または機能に対して行われた高いレベルの変更が記録されます。このような変更を行ったユーザも記録されます。

情報メッセージ（6：情報）のロギングをイネーブルにすると、RDU はバッチ処理操作を公開する追加のメッセージを記録します。これらのメッセージには、経過時間とレートに関する情報も含まれます。

rdu.log ファイルの表示

rdu.log ファイルを表示するには、任意のテキスト エディタを使用できます。また、このログ ファイルは、管理者のユーザ インターフェイスからも表示できます。

ファイルを表示するには、次の手順に従います。

ステップ 1　**Servers** の下の RDU タブを選択します。

View Regional Distribution Unit Details ページが表示されます。

ステップ 2　RDU Log File に対応する View Details アイコン（）をクリックします。

View Log File Contents ページが表示され、*rdu.log* からのデータが示されます。

audit.log ファイルの表示

audit.log ファイルを表示するには、任意のテキスト エディタを使用できます。また、このログ ファイルは、管理者のユーザ インターフェイスからも表示できます。

ファイルを表示するには、次の手順に従います。

ステップ 1　**Servers** の下の RDU タブを選択します。

View Regional Distribution Unit Details ページが表示されます。

ステップ 2　Audit Log File に対応する View Details アイコンをクリックします。

View Log File Contents ページが表示され、*audit.log* からのデータが示されます。

RDU ログ レベル ツールの使用方法

RDU ログ レベル ツールを使用して、コマンドラインから RDU の現在のログ レベルを変更します。**setLogLevel.sh** コマンドを使用します。このツールは *BPR_HOME/rdu/bin* ディレクトリにあります。

表 10-2 に利用可能な重大度のレベルとログがイネーブルになっているときにログ ファイルに書き込まれるメッセージのタイプを示します。

表 10-2　ログ レベル

ログ レベル	説明
0：緊急	システムが不安定です。すべての緊急メッセージを保存するように、ロギング機能を設定します。
1：アラート	すぐに対応が必要です。すぐに対応が必要なすべてのアクティビティ、およびさらに深刻な活動を保存するように、ロギング機能を設定します。
2：クリティカル	クリティカルな状態が存在します。すべてのエラー メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
3：エラー	エラー状態が存在します。すべてのエラー メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
4：警告	警告状態が存在します。すべての警告メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
5：通知	通常ですが、重大な状態が存在します。すべての通知メッセージ、およびさらに深刻なメッセージを保存するように、ロギング機能を設定します。
6：情報	情報メッセージ。利用可能なすべてのロギング メッセージを保存するように、ロギング機能を設定します。

（注）　7（デバッグ）として知られるもう 1 つのレベルは、シスコでデバッグの目的にのみ使用されます。Cisco TAC で指示された場合を除き、このレベルは使用しないようにしてください。

安定した動作状態を維持するためには、RDU 重大度のレベルを警告レベルのままにすることをお勧めします。デバッグ動作中に安定した状態パフォーマンスを維持する必要がある場合は、情報レベルを注意して使用することをお勧めします。情報レベルで実行すると大量のログ エントリが作成され、このことがパフォーマンスに悪影響を与える可能性があるため、注意が必要です。

（注）　ログ レベル ツールを実行するには、RDU プロセスが稼働している必要があります。また、**setLogLevel.sh** コマンドを使用してこのツールを実行する特権も必要です。

シンタックスの説明　**setLogLevel.sh** -[*0..6*] [**-help**] [**-show**] [**-default**] [**-debug**]

- -[*0..6*]：使用する重大度のレベルを示します。利用可能なレベルのリストについては、表 10-2 を参照してください。
- **-help**：ツールのヘルプを表示します。
- **-show**：RDU サーバの現在の重大度のレベル設定を表示します。
- **-default**：RDU をインストール デフォルト レベルの 5（通知）に設定します。
- **-debug**：RDU サーバのカテゴリのトレースをイネーブルまたはディセーブルにするように、対話モードを設定します。

（注）　シスコのサポート スタッフの指示があった場合にのみ、デバッグ設定をイネーブルにしてください。

このツールを使用して、次の機能も実行できます。

- RDU ログ レベルの設定（P.10-6）
- RDU の現在のログ レベルの表示（P.10-7）

RDU ログ レベルの設定

このツールを使用して、ロギング レベルをある値から別の値に変更できます。次の例では、RDU ロギング レベルを警告レベル（**setLogLevel.sh** コマンドでは数値 4 で示されるレベル）に設定する方法を示します。実際のログ レベル設定は手順にとって重要ではないので、必要に応じて読み替えてください。

この項で示している例は、RDU サーバが稼動中で、RDU のユーザ名が **admin**、パスワードが **changeme** であることを想定しています。

RDU ロギング レベルを設定するには、次の手順に従います。

ステップ 1　*BPR_HOME/rdu/bin* にディレクトリを変更します。

ステップ 2　次のコマンドを使用して、RDU ログ レベル ツールを実行します。

```
# setLogLevel.sh 4
```

次のプロンプトが表示されます。

```
Please type RDU username:
```

ステップ 3　RDU ユーザ名を入力します。この例では、デフォルト ユーザ名（**admin**）を使用します。

```
Please type RDU username: admin
```

次のプロンプトが表示されます。

```
Please type RDU password:
```

ステップ 4　RDU の RDU パスワードを入力します。この例では、デフォルト パスワード（**changeme**）を使用します。

```
Please type RDU password: changeme
```

次のメッセージが表示され、ログ レベルが変更されたことが通知されます。この例では、レベル 5（通知）から 4（警告）に変更されました。

```
RDU Log level was changed from 5 (notification) to 4 (warning).
```

RDU の現在のログ レベルの表示

このツールを使用して、ロギング レベルの値を変更する前に、RDU ログを表示し、設定されている値を判別できます。

この項で示している例は、次のことを前提としています。

- RDU サーバが稼動中である。
- RDU のユーザ名が **admin** である。
- パスワードが **changeme** である。

RDU の現在のロギング レベルを表示するには、次の手順に従います。

ステップ 1　*BPR_HOME/rdu/bin* にディレクトリを変更します。

ステップ 2　次のコマンドを実行します。

```
# setLogLevel.sh -show
```

次のプロンプトが表示されます。

```
Please type RDU username:
```

ステップ 3　RDU ユーザ名（**admin**）を入力し、**Enter** キーを押します。

```
Please type RDU username: admin
```

次のプロンプトが表示されます。

```
Please type RDU password:
```

ステップ 4　RDU パスワード（**changeme**）を入力し、**Enter** キーを押します。

```
Please type RDU password: changeme
```

次のメッセージが表示されます。

```
The logging is currently set at level: 4 (warning)
All tracing is currently disabled.
```

DPE のログ

DPE は *BPR_DATA/dpe/logs* ディレクトリに *dpe.log* ファイルを保持しています。このファイルには、デフォルト レベルが設定されているすべてのイベントのレコードが含まれます。システム障害が連続して起こるなど、DPE で破局的な障害が発生した場合、破局的なエラーは *rdu.log* ファイルにも記録されます。

SNMPService.logyyy.log ログ ファイルは、DPE サーバで PacketCable がイネーブルになっているときに、詳細なデバッグ情報を提供するために DPE が使用します。DPE コマンドライン インターフェイス（CLI）から **service packetcable** *1..1* **show snmp log** コマンドを使用して、このファイルを表示します。このファイルは *BPR_DATA/dpe/logs* ディレクトリにあります。PacketCable コマンドの使用方法については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

（注） PacketCable ロギング メッセージは *dpe.log* ファイルに送信され、詳細な SNMP デバッグは *SNMPService.logyyy.log* ファイルに送信されます。

dpe.log ファイルを表示するには、任意のテキスト ビューアを使用できます。また、DPE CLI から **show log** コマンドを使用することもできます。詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

さらに、BAC 管理者のユーザ インターフェイスを使用して DPE ログ ファイルを表示することもできます。

ファイルを表示するには、次の手順に従います。

ステップ 1　**Servers** > **DPEs** を選択します。

ステップ 2　ログ ファイルを表示する DPE のリンクをクリックします。

View Device Provisioning Engines Details ページが表示されます。

ステップ 3　*dpe.log* ファイルの内容を表示するには、Log Files 領域で DPE Log File の View Details アイコンをクリックします。

Network Registrar のログ

BAC は Cisco Network Registrar DHCP サーバ拡張からログ メッセージを生成します。DHCP サーバのログは *cnr-install-path/name_dhcp_1_log* ディレクトリにあります。*cnr-install-path* は変数で、入力する値に固有のものです。DHCP サーバのログ ファイルのデフォルトの場所は */var/nwreg2/local/logs/name_dhcp_1_log* です。

DHCP サーバ拡張経由で出力されるログ メッセージは、拡張トレース レベルの設定に基づいています。トレース レベルで値を設定できます（表 10-3 を参照）。設定する数値が、すべての拡張の **extension-trace-level** 属性の現在の設定の数値になります。

表 10-3　DHCP サーバ拡張トレース レベル

レベル	説明
0	エラーと警告の状態を記録します。すべてのエラーと警告メッセージ、およびさらに深刻なメッセージを出力するように、拡張を設定します。
1	サーバ インタラクションを記録します。DPE から取得する構成命令と RDU に転送される構成生成要求が含まれます。
2	処理の詳細を記録します。命令生成要求に転送された個々の設定コマンドと属性値が含まれます。
3	拡張デバッグの内部処理を記録します。これには、メッセージの 16 進ダンプが含まれます。
4	拡張バックグラウンド操作のデバッグを記録します。これには、DPE 状態のポーリングが含まれます。

拡張トレース レベルは、Network Registrar Web UI を使用して変更できます。レベルを変更するには、次の手順に従います。

ステップ 1　Network Registrar のローカル Web UI を開きます。

ステップ 2　メニューから、**DHCP**、**DHCP Server** の順にクリックします。

ステップ 3　Local DHCP Server リンクをクリックします。

ステップ 4　Edit DHCP Server ページで、Extensions 属性カテゴリを展開します。

ステップ 5　**extension-trace-level** 値を設定して、**Modify Server** をクリックします。

ステップ 6　DHCP サーバをリロードします。

（注）　DHCP サーバで実行されるロギングの詳細については、『*User Guide for Cisco Network Registrar 7.0*』を参照してください。

SNMP の使用によるサーバの監視

BAC では、SNMP を使用したサーバの監視がサポートされています。具体的には、SNMP ベースの管理システムを使用して、BAC サーバの状態、ライセンスの使用状況情報、サーバ接続、およびサーバ固有の統計情報を監視できます。

SNMP エージェント

BAC では、RDU サーバおよび DPE サーバについて基本的な SNMP v2 ベースの監視がサポートされます。BAC SNMP エージェントでは SNMP 通知と SNMP トラップがサポートされます。それらをまとめて「通知」と呼びます。snmp-server CLI コマンドを使用して DPE に SNMP エージェントを設定し、SNMP 設定コマンドライン ツールを使用して RDU に SNMP エージェントを設定できます。

SNMP 設定コマンドライン ツールの詳細については、P.10-11 の「snmpAgentCfgUtil.sh ツールの使用方法」を参照してください。DPE CLI の詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

MIB のサポート

BAC では、数種類の MIB がサポートされます。表 10-4 に、各 BAC コンポーネントに対する MIB サポートの概要を示します。

表 10-4　BAC でサポートされる MIB

コンポーネント	サポート対象の MIB
DPE	CISCO-BACC-SERVER-MIB
	CISCO-BACC-DPE-MIB
RDU	CISCO-BACC-SERVER-MIB
	CISCO-BACC-RDU-MIB

SNMP エージェントは CISCO-BACC-SERVER-MIB をサポートします。この MIB は、BAC 上のすべてのサーバに共通の管理対象オブジェクトを定義します。この MIB は、同一のデバイスにインストールされている複数の BAC サーバの監視をサポートします。サーバの状態が変化するたびに `ciscoBaccServerStateChanged` 通知が生成されます。

RDU SNMP エージェントでは、RDU の管理対象オブジェクトを定義する CISCO-BACC-RDU-MIB がサポートされます。この MIB は、RDU の状態に関する統計情報、RDU と DPE の間および RDU と Network Registrar の間の通信インターフェイスに関する統計情報を定義します。

SNMP エージェントは `cnaHealthNotif` トラップを生成します。このトラップは、RDU サーバの起動、シャットダウン、障害、または終了ステータスの変化を通知します。

DPE SNMP エージェントでは、DPE にインストールされているコンポーネントの管理対象オブジェクトを定義する CISCO-BACC-DPE-MIB がサポートされます。DPE は、デバイス構成のローカルキャッシング、およびサポートされているすべてのデバイスで使用される設定ファイルを管理します。この MIB は、TFTP サーバと ToD サーバのエントリを含む、基本的な DPE 設定情報と統計情報を提供します。

SNMP エージェントは CISCO-NMS-APPL-HEALTH-MIB もサポートします。この MIB は、Cisco NMS アプリケーションのヘルス ステータスの通知と関連オブジェクトを定義します。これらの通知は、NMS アプリケーションのステータス（起動、停止、失敗、ビジー、またはアプリケーションの異常終了）を知らせるために OSS/NMS に送信されます。デフォルトの MIB は MIB-II です。

（注）　すべてのオブジェクトの説明については、*BPR_HOME/rdu/mibs* ディレクトリにある対応する MIB ファイルを参照してください。

snmpAgentCfgUtil.sh ツールの使用方法

snmpAgentCfgUtil.sh ツールを使用すると、Solaris コンピュータにインストールされている SNMP エージェントを管理できます。このツールは *BPR_HOME/snmp/bin* ディレクトリにあり、これを使用して、SNMP 通知を受信する他のホストのリストにホストを追加（またはリストから削除）し、SNMP エージェント プロセスを起動および中止できます。このツールはローカル ディレクトリから実行する必要があります。

（注）　Solaris コンピュータ上で動作する SNMP エージェントのデフォルト ポート番号は 8001 です。

RDU SNMP エージェントは、次の操作に使用できます。

- ホストの追加（P.10-11）
- ホストの削除（P.10-12）
- SNMP エージェント コミュニティの追加（P.10-12）
- SNMP エージェント コミュニティの削除（P.10-13）
- SNMP エージェントの開始（P.10-13）
- SNMP エージェントの停止（P.10-14）
- SNMP エージェント リスニング ポートの設定（P.10-14）
- SNMP エージェントの場所の変更（P.10-14）
- SNMP の連絡先の設定（P.10-15）
- SNMP エージェントの設定の表示（P.10-15）
- SNMP 通知タイプの指定（P.10-15）

ホストの追加

SNMP エージェントから SNMP 通知を受信するホストのリストにホスト アドレスを追加するには、次のコマンドを使用します。

シンタックスの説明　***snmpAgentCfgUtil.sh add host*** *ip-addr* ***community*** *community* [***udp-port*** *port*]

- *ip-addr*：通知を送信するホストの IP アドレスを指定します。
- *community*：SNMP 通知を送信するときに使用するコミュニティ（読み取りまたは書き込み）を指定します。
- *port*：SNMP 通知の送信に使用する UDP ポートを示します。

例

```
# ./snmpAgentCfgUtil.sh add host 10.10.10.5 community trapCommunity udp-port 162
OK
Please restart [stop and start] SNMP agent.
```

（注）　このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

ホストの削除

SNMP エージェントから SNMP 通知を受信するホストのリストからホストを削除するには、次のコマンドを使用します。

シンタックスの説明　***snmpAgentCfgUtil.sh delete host*** *ip-addr*

ip-addr：ホストのリストから削除するホストの IP アドレスを指定します。

例

```
# ./snmpAgentCfgUtil.sh delete host 10.10.10.5
OK
Please restart [stop and start] SNMP agent.
```

（注）　このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP エージェント コミュニティの追加

SNMP コミュニティ ストリングを追加して、SNMP エージェントへのアクセスを許可するには、次のコマンドを使用します。

シンタックスの説明　***snmpAgentCfgUtil.sh add community*** *string* [***ro*** | ***rw***]

- *string*：SNMP コミュニティを示します。
- ***ro***：読み取り専用（**ro**）のコミュニティ ストリングを割り当てます。実行できるのは 取得要求（クエリー）だけです。ro コミュニティ ストリングは、取得要求を許可しますが、設定操作は許可しません。NMS と管理対象デバイスは、同じコミュニティ ストリングを参照する必要があります。
- **rw**：読み取りと書き込み（**rw**）コミュニティ ストリングを割り当てます。SNMP アプリケーションでは、設定操作に rw アクセスが必要です。**rw** コミュニティ ストリングを使用すると、OID 値への書き込みアクセスがイネーブルになります。

（注）　デフォルトの **ro** および **rw** コミュニティ ストリングは、それぞれ `baccread` と `baccwrite` です。BAC を配備する前に、これらの値を変更することをお勧めします。

例

```
# ./snmpAgentCfgUtil.sh add community fsda54 ro
OK
Please restart [stop and start] SNMP agent.
```

（注） このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP エージェント コミュニティの削除

SNMP コミュニティ ストリングを削除して、SNMP エージェントへのアクセスを禁止するには、次のコマンドを使用します。

シンタックスの説明　***snmpAgentCfgUtil.sh delete community*** *string* [***ro*** | ***rw***]

- *string*：SNMP コミュニティを示します。
- ***ro***：指定したコミュニティが読み取り専用コミュニティであることを示します。
- ***rw***：指定したコミュニティが読み取りと書き込みコミュニティであることを示します。

（注） **ro** および **rw** コミュニティ ストリングの詳細については、P.10-12 の「SNMP エージェント コミュニティの追加」を参照してください。

例

```
# ./snmpAgentCfgUtil.sh delete community fsda54 ro
OK
Please restart [stop and start] SNMP agent.
```

（注） このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP エージェントの開始

BAC がインストールされている Solaris コンピュータで SNMP エージェント プロセスを開始するには、次のコマンドを使用します。

（注） SNMP エージェントは、**/etc/init.d/bprAgent start snmpAgent** コマンドを使用して BAC プロセス ウォッチドッグを起動することでも開始できます。詳細については、P.9-2 の「コマンドラインからの BAC プロセス ウォッチドッグの使用方法」を参照してください。

例

```
# ./snmpAgentCfgUtil.sh start
Process snmpAgent has been started
```

SNMP エージェントの停止

BAC がインストールされている Solaris コンピュータで SNMP エージェント プロセスを停止するには、次のコマンドを使用します。

（注） SNMP エージェントは、**/etc/init.d/bprAgent stop snmpAgent** コマンドを使用して BAC プロセス ウォッチドッグを起動することでも停止できます。詳細については、P.9-2 の「コマンドラインからの BAC プロセス ウォッチドッグの使用方法」を参照してください。

例

```
# ./snmpAgentCfgUtil.sh stop
Process snmpAgent has stopped
```

SNMP エージェント リスニング ポートの設定

SNMP エージェントがリッスンするポート番号を指定するには、次のコマンドを使用します。RDU SNMP エージェントが使用するデフォルト ポート番号は 8001 です。

シンタックスの説明

snmpAgentCfgUtil.sh udp-port *port*

port：SNMP エージェントがリッスンするポート番号を示します。

例

```
# ./snmpAgentCfgUtil.sh udp-port 8001
OK
Please restart [stop and start] SNMP agent.
```

（注） このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP エージェントの場所の変更

SNMP エージェントを実行するデバイスの場所を示すテキスト文字列を入力するには、次のコマンドを使用します。たとえば、このコマンドを使用してデバイスの物理的な場所を示すことができます。最大 255 文字の任意の文字列を入力できます。

シンタックスの説明

snmpAgentCfgUtil.sh location *location*

location は、エージェントの場所を示す文字列です。

例

次の例では、SNMP エージェントの物理的な場所は、rack 5D と示された装置ラックです。

```
# ./snmpAgentCfgUtil.sh location "equipmentrack5D"
OK
Please restart [stop and start] SNMP agent.
```

（注）　このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP の連絡先の設定

SNMP エージェントの連絡担当者と、この担当者への連絡方法を示すテキスト文字列を入力するには、次のコマンドを使用します。たとえば、このコマンドを使用して、特定の担当者（電話番号を含む）を示すことができます。最大 255 文字の任意の文字列を入力できます。

シンタックスの説明　***snmpAgentCfgUtil.sh contact*** *contact-info*

contact-info は、SNMP エージェントに関する連絡担当者を示す文字列です。

例　次の例では、連絡担当者の名前は Terry で、内線番号は 1234 です。

```
# ./snmpAgentCfgUtil.sh contact "Terry-ext1234"
OK
Please restart [stop and start] SNMP agent.
```

（注）　このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

SNMP エージェントの設定の表示

現在の SNMP 設定をすべて表示するには、次のコマンドを使用します。

例

```
# ./snmpAgentCfgUtil.sh show
Location                        : equipmentrack5D
Contact                         : Terry-ext1234
Port Number                     : 8001
Notification Type               : trap
Notification Recipient Table    :
        [ Host IP address, Community, UDP Port ]
        [ 10.10.10.5 , trapCommunity , 162 ]
Access Control Table            :
        Read Only Communities
                baccread
        Read Write Communities
                baccwrite
```

SNMP 通知タイプの指定

SNMP エージェントから送信される通知のタイプ（トラップまたは通知）を指定するには、次のコマンドを使用します。デフォルトではトラップが送信されますが、SNMP 通知を送信するようにエージェントを設定することもできます。

シンタックスの説明　***snmpAgentCfgUtil.sh inform*** [***retries timeout***] | ***trap***

パラメータは、リトライ間のバックオフ タイムアウトです。

例

```
# ./snmpAgentCfgUtil.sh inform retries 3 timeout 1000
OK
Please restart [stop and start] SNMP agent.
```

（注） このコマンドを使用して加えた変更は、**/etc/init.d/bprAgent restart snmpAgent** コマンドを使用して SNMP エージェントを再起動するまで有効になりません。詳細については、P.9-2 の「BAC プロセス ウォッチドッグ」を参照してください。

設定内容を確認するには、**snmpAgentCfgUtil.sh show** コマンドを使用します。

```
# ./snmpAgentCfgUtil.sh show
Location                        : equipmentrack5D
Contact                         : Terry-ext1234
Port Number                     : 8001
Notification Type               : inform
Notification Retries            : 3
Notification Timeout            : 1000
Notification Recipient Table    :
        [ Host IP address, Community, UDP Port ]
        [ 10.10.10.5 , trapCommunity , 162 ]
Access Control Table            :
        Read Only Communities
                baccread
        Read Write Communities
                baccwrite
```

サーバ状態の監視

この項では、BAC 配備内の RDU サーバおよび DPE サーバのパフォーマンスを監視する方法について説明します。監視対象のサーバは、中央 RDU サーバと DPE サーバです。

サーバ統計情報は、次の手段で確認できます。

- 管理者のユーザ インターフェイス
- DPE CLI
- RDU ログ ファイルおよび DPE ログ ファイル（管理者のユーザ インターフェイスまたは DPE CLI を使用）

管理者のユーザ インターフェイスの使用方法

管理者のユーザ インターフェイスで利用可能なサーバ統計情報を表示するには、次の手順に従います。

1. プライマリ ナビゲーション バーの **Server** タブをクリックします。

   セカンダリ ナビゲーション バーに、DPEs、NRs、Provisioning Group、RDU といったオプションが表示されます。

2. 次のいずれかをクリックします。
   - DPEs タブ：BAC データベースに現在登録されているすべての DPE を監視する場合
   - RDU タブ：RDU の状態および統計情報を表示する場合

   クリックしたタブに応じて、次のように表示されます。
   - DPEs：Manage Device Provisioning Engine ページが表示されます。

     このページに表示される各 DPE 名は、その DPE の詳細を表示する別ページへのリンクになっています。詳細ページを表示するには、このリンクをクリックします。
   - RDU：View Regional Distribution Unit Details ページが表示されます。

DPE CLI の使用方法

DPE サーバの状態を監視するには、**show dpe** コマンドを実行して、DPE が動作しているかどうかを確認し、プロセスの状態と、DPE が動作している場合は、動作状態に関する統計情報を表示します。例 10-2 を参照してください。

（注） このコマンドでは、DPE が正常に動作しているかどうかは示されません。プロセス自体が現在実行されていることだけが示されます。ただし、DPE が動作していれば、このコマンドで出力される統計情報を使用して、DPE が正常に要求を処理しているかどうかを判別できます。

例 10-2　show dpe の出力

この結果は、DPE が動作している場合に発生します。

```
dpe# show dpe
BAC Agent is running
Process dpe is running
Version BAC 4.0 (SOL_CBAC4_0_L_000000000000).
Caching 1 device configs and 1 external files.
0 sessions succeed and 0 sessions failed.
0 file requests succeed and 0 file requests failed.
0 immediate proxy operations received: 0 succeed, and 0 failed.
Connection status is Ready.
Running for 4 hours 30 mins 16 secs.
```

この結果は、DPE が動作していない場合に発生します。

```
dpe_host# show dpe
BAC Agent is running
Process dpe is not running
```

（注） 詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

CHAPTER 11

管理者のユーザ インターフェイスについて

この章では、Broadband Access Center（BAC）管理者のユーザ インターフェイスにアクセスする方法と、そのインターフェイス自体について説明します。この章では次のトピックについて説明します。

管理者のユーザ インターフェイスの設定

管理者のユーザ インターフェイスを使用する前に、*adminui.properties* ファイルの内容を確認してください。このファイルには、インターフェイスの動作を指定する各種のコントロールが含まれています。

テキスト エディタを使用してこのファイルを開き、必要な機能を実行するようにその内容を変更できます。変更を保存後、ユーザ インターフェイスを再起動すると、変更内容が有効になります。

管理者のユーザ インターフェイスを起動するには、次のように入力します。

```
# /etc/init.d/bprAgent start tomcat
```

管理者のユーザ インターフェイスを停止するには、次のように入力します。

```
# /etc/init.d/bprAgent stop tomcat
```

管理者のユーザ インターフェイスを再起動するには、次のように入力します。

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

このユーザ インターフェイスは、*adminui.properties* ファイルにあるオプションを使用して設定できます。これらのオプションは、BAC の設定によって制御するか、*BPR_HOME/rdu/conf* ディレクトリにある *adminui.properties* ファイルで定義します。構成パラメータは次のとおりです。

- `/adminui/port`：RDU のリスニング ポートを指定します。デフォルトのポート番号は 49187 です。
- `/adminui/fqdn`：RDU を実行するホストの完全修飾ドメイン名を指定します。デフォルト値はホストの FQDN です。たとえば、bac_test.EXAMPLE.COM です。
- `/adminui/maxReturned`：検索結果の最大件数を指定します。デフォルト値は 1000 です。

- `/adminui/pageSize`：1 ページに表示される検索結果の件数を指定します。この数は 25、50、または 75 に設定できます。デフォルト値は 25 です。
- `/adminui/refresh`：リフレッシュ機能をイネーブルにするか、ディセーブルにするかを指定します。デフォルトでは、このオプションはディセーブルになっています。
- `/adminui/extensions`：BAC での拡張の使用をイネーブルにするか、ディセーブルにするかを指定します。拡張を使用すると、BAC の動作を強化したり、新しいデバイス テクノロジーのサポートを追加したりできます。デフォルトでは、拡張の使用はイネーブルになっています。
- `/adminui/maxFileSize`：BAC　にアップロードされるファイルの最大サイズを指定します。デフォルトのファイル サイズは 4 MB です。
- `/adminui/refreshRate`：画面がリフレッシュされるまでの時間の長さ（秒数）を指定します。デフォルト値は 90 秒です。このオプションの値を設定する前に、`/adminui/refresh` オプションがイネーブルであることを確認してください。
- `/adminui/file/extensions`：ユーザ インターフェイスがサポートするファイルの拡張子を指定します。デフォルトでサポートされている拡張子は、*.bin*、*.cm*、および *.jar* です。
- `/adminui/timeout`：アイドル セッションがタイムアウトになるまでの時間の長さ（秒数）を指定します。デフォルトでは、この期間は 300 秒に設定されます。
- `/adminui/noOfLines`：このユーザ インターフェイスに表示される *rdu.log* または *dpe.log* の最後の行番号を指定します。デフォルトでは、表示される行数は 250 です。

例 11-1　adminui.properties ファイルのサンプル

```
/adminui/port=49187
/adminui/fqdn=doc.example.com
/adminui/maxReturned=1000
/adminui/pageSize=25
/adminui/refresh=disabled
/adminui/extensions=enabled
/adminui/maxFileSize=40000000
/adminui/refreshRate=90
/adminui/timeout=300
/adminui/noOfLines=250
```

管理者のユーザ インターフェイスへのアクセス

BAC ユーザ インターフェイスには、BAC アプリケーションの URL にアクセスできる任意のコンピュータからアクセスできます。

ログイン

BAC インターフェイスには、管理ユーザ、読み取り / 書き込みユーザ、または読み取り専用ユーザとしてログインできます。P.12-2 の「ユーザ管理」で説明するとおり、実行可能な機能は各ユーザタイプで異なりますが、ユーザ インターフェイスには同じ方法でアクセスします。

BAC 管理者のユーザ インターフェイスにアクセスするには、次の手順に従います。

ステップ 1　Web ブラウザを起動します。

表 11-1 は、この BAC リリースでサポートされるブラウザのリストを示しています。

表 11-1　ブラウザのプラットフォーム サポート

プラットフォーム	サポートするブラウザ
Windows 2000（Service Pack 2）	Internet Explorer 6.0 以降 Firefox 1.5 以降
Red Hat Linux (7.1)	Firefox 1.5 以降
Solaris (2.9)	Firefox 1.5 以降

ステップ 2　次の構文を使用して、管理者の場所を入力します。

`http://machine_name:port_number/`

- *machine_name*：RDU を実行しているコンピュータを示します。

（注） HTTP over SSL（HTTPS とも呼ばれる）を使用して管理者のユーザ インターフェイスにアクセスするには、`https://machine_name:port_number/` と入力します。

- *port_number*：サーバ側の管理アプリケーションを実行するコンピュータ ポートを示します。デフォルトでは、このポート番号は次のとおりです。
  - HTTP over TCP の場合：8100
  - HTTP over SSL の場合：8443

メイン ログイン ページ（図 11-1）が表示されます。

図 11-1　ログイン ページ

ステップ 3　デフォルトのユーザ名（**admin**）とデフォルトのパスワード（**changeme**）を入力します。

初めてログインする場合は、Change Password 画面（図 11-2）が表示されます。

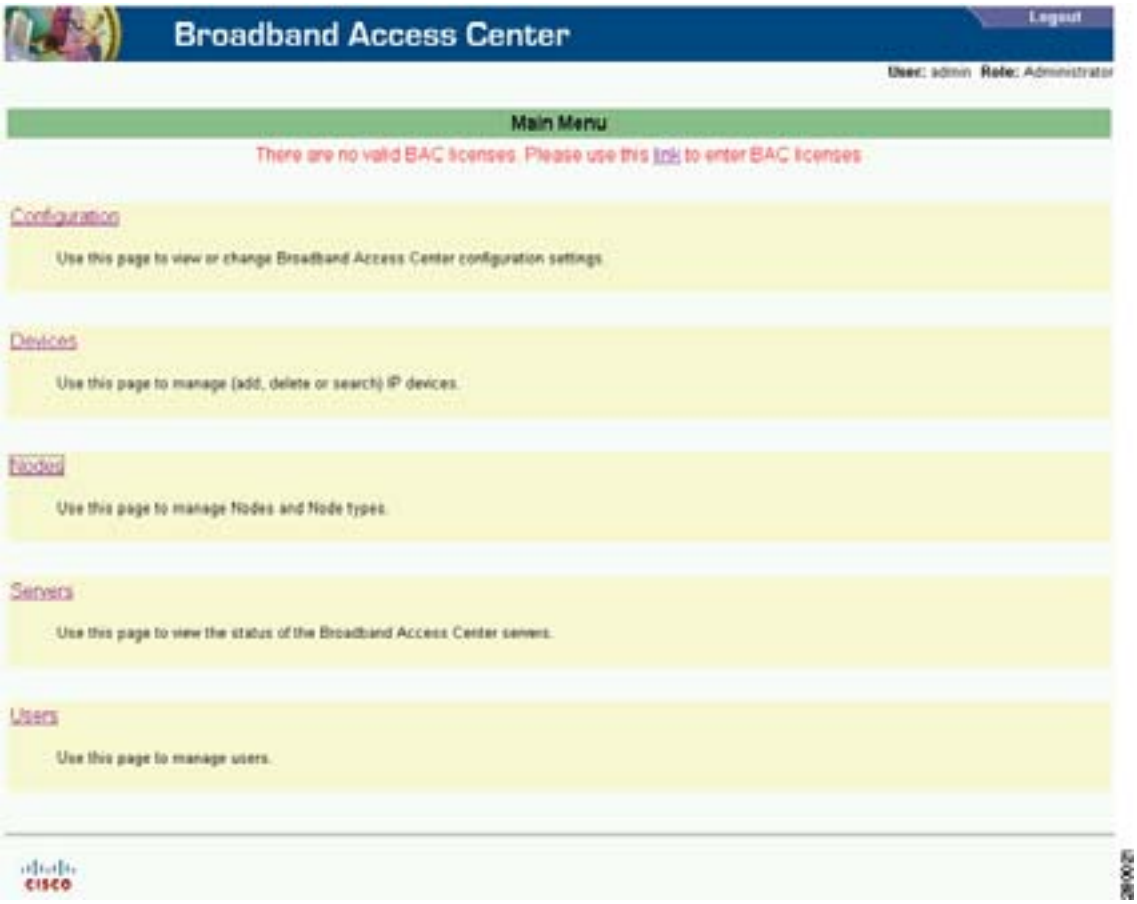
図 11-2　Change Password 画面

新しいパスワードを入力して、確認します。8 文字以上のパスワードを入力してください。

ステップ 4　**Login** をクリックします。

Main Menu ページ（図 11-3）が表示されます。

図 11-3　Main Menu ページ



（注）　ページの上部にあるリンクを使用して、ライセンス　ファイルを追加できます。詳細については、P.13-23 の「ライセンスの管理」を参照してください。

ログアウト

BAC からログアウトするには、次の手順に従います。

ステップ 1　ページの右上にある **Logout** タブをクリックします。

確認ダイアログボックスが表示されます。

ステップ 2　**OK** をクリックします。

アプリケーションにより、Login ページに戻ります。図 11-1 を参照してください。

管理者のユーザ インターフェイスのアイコンについて

BAC 管理者のユーザ インターフェイスには、特定の機能を実行するときに使用できるアイコンがあります。表 11-2 は、これらのアイコンを示しています。

表 11-2　管理者のユーザ インターフェイスのアイコン

アイコン	説明
	View Details アイコン：特定のデバイスまたはファイルの詳細を表示できます。
	Delete アイコン：特定のオブジェクトを削除します。
	Export アイコン：特定のファイルの内容をクライアント コンピュータにエクスポートします。

これらのアイコンは、管理者のユーザ インターフェイスで実行する手順に関する項で使用されています。これらの項は、以下に記載されています。

- 第 12 章「管理者のユーザ インターフェイスの使用方法」
- 第 13 章「Broadband Access Center の設定」

CHAPTER 12

管理者のユーザ インターフェイスの使用方法

この章では、Broadband Access Center（BAC）管理者のユーザ インターフェイスから実行する管理作業について説明します。管理作業には主に、次のような BAC コンポーネントのアクションの監視があります。

（注） この章で説明する手順は、チュートリアルの形式で示されています。可能な限り、各手順の結果を表す例を示すようにしてあります。

ユーザ管理

ユーザの管理には、BAC を管理するユーザの追加、修正、削除があります。ユーザ タイプによっては、このメニューを使用して、ユーザを追加、修正、および削除できます。このメニューには、BAC を使用するように設定されているユーザがすべて表示され、それらのユーザのユーザ タイプも示されます。

BAC ユーザには、管理者、読み取り / 書き込みユーザ、読み取り専用ユーザという 3 つのタイプがあります。各ユーザ タイプはアクセス レベルが異なり、一意のアクセス権を付与されているため、アクセスを確実に制御してプロビジョニング データの一貫性を保つことができます。

割り当てられているユーザ タイプは、管理者のユーザ インターフェイスの各画面の右上近くに表示されます。

管理者

BAC が認識する管理者は 1 名のみです。このユーザは、デバイス データを表示、追加、修正、削除したり、他のユーザを作成したりできます。管理者は、他のユーザのアクセス権を読み取り / 書き込みから読み取り専用に変更することや、読み取り専用から読み取り / 書き込みに変更することもできます。また、他の任意のユーザ タイプのパスワードを変更することもできます。

管理者ユーザを削除することはできません。

読み取り / 書き込みユーザ

読み取り / 書き込みユーザは管理者と同じ機能を実行できますが、他のユーザを作成することや、他のユーザのユーザ タイプおよびパスワードを変更することはできません。読み取り / 書き込みユーザは、自分のパスワードを変更できます。

読み取り専用ユーザ

読み取り専用ユーザは、自分のパスワードの変更や、デバイス データの表示などの基本的なアクセスを実行できますが、デバイス データを変更することはできません。動作が中断されると見なされる操作は一切実行できません。たとえば、構成のリセットや再生成は実行できません。

（注） 許容される以前のリリースから BAC 4.0 へ移行する間に、すべての移行済みユーザに読み取り / 書き込み特権が割り当てられます。

ユーザを追加または削除できるのは、管理者としてログインしている場合のみです。

次の各項では、BAC ユーザ管理の次の手順について説明します。

- 新規ユーザの追加
- ユーザの修正
- ユーザの削除

新規ユーザの追加

新規ユーザの追加は、ユーザ名を入力し、パスワードを作成する単純な手順です。ただし、新規ユーザを作成するときは、ユーザを読み取り / 書き込みユーザと読み取り専用ユーザのどちらのタイプにするかを決定する必要があります。

（注） BAC には、1 つの**管理者**ユーザがあらかじめ作成されています。管理者を新規ユーザとして作成することはできません。

新規ユーザを追加するには、次の手順に従います。

ステップ 1　**Users** タブをクリックします。

Manage Users ページが表示されます

ステップ 2　**Add** をクリックして Add User ページを表示します。

ステップ 3　新規ユーザのユーザ名を入力します。

ステップ 4　パスワードを入力して、確認します。8 文字以上のパスワードを入力してください。

ステップ 5　該当するオプション ボタンをクリックして新規ユーザのロールを決定します。各ユーザ タイプの詳細については、前の各項を参照してください。

ステップ 6　新規ユーザの簡単な説明を入力します。

ヒント　説明フィールドを使用して、ユーザの仕事、役職、またはその新規ユーザを特定する固有の情報を指定します。

ステップ 7　**Submit** をクリックします。

新規ユーザが追加された状態で Manage Users ページが表示されます。

（注） 紛失や盗難、あるいは無権限でのアクセスを防止するため、新規ユーザのパスワードは、必ず記録して安全な場所に保管してください。

ユーザの修正

どのタイプのユーザでも自分のパスワードとユーザ説明は修正できますが、別のユーザの情報を修正できるのは管理者のみです。

ユーザ プロパティを変更するには、次の手順に従います。

ステップ 1　**Users** タブをクリックします。

Manage User ページが表示されます。

ステップ 2　適切なユーザ名をクリックして、そのユーザの Modify User ページにアクセスします。

ステップ 3　パスワード、ユーザ タイプ（管理者としてログインしていることが必須）、およびユーザ説明に対して必要な変更を行います。

ステップ 4　**Submit** をクリックします。

ユーザ情報が変更された状態で Manage Users ページが表示されます。

ユーザの削除

Manage Users ページに表示される管理者以外のユーザは、管理者のみが削除できます。**admin** という名前のデフォルト ユーザを削除することはできません。ユーザを削除するには、次の手順に従います。

ステップ 1　**Users** をクリックします。

Manage User ページが表示されます。

ステップ 2　削除するユーザに対応する **Delete** アイコン（🗑）をクリックします。

Delete User ダイアログボックスが表示されます。

ステップ 3　**OK** をクリックします。

削除したユーザが含まれていない状態で Manage Users ページが表示されます。

デバイス管理

Devices メニューを使用してさまざまなデバイスをプロビジョニングおよび管理します。次の操作を実行できます。

- 特定のデバイス、または指定した基準を共有しているデバイスのグループを検索する。P.12-5 の「デバイスの検索」を参照してください。
- RDU データベースを対象として、デバイスを追加、修正、または削除する。次の各項を参照してください。
  - デバイス レコードの追加（P.12-14）
  - デバイス レコードの修正（P.12-15）
  - デバイスの削除（P.12-15）
- 設定やプロパティなどのデバイス データを表示する。P.12-10 の「デバイスの詳細の表示」を参照してください。
- デバイス構成を再生成する。P.12-16 の「デバイス構成の再生成」を参照してください。
- 任意のデバイスを、特定のノードに関連付けまたは関連付け解除する。P.12-17 の「デバイスの関連付けと関連付け解除」を参照してください。
- デバイスをリセットまたはリブートする。P.12-18 の「デバイスのリセット」を参照してください。

Manage Devices ページ

Manage Devices ページは、プライマリ ナビゲーション バーの **Devices** タブをクリックすると表示されます。Main Menu の Devices リンクをクリックして Manage Devices ページに移動することもできます。

デバイスの検索

BAC を使用して、さまざまな方法でデバイス情報を検索できます。

検索タイプを選択するには、Manage Devices ページで Search Type ドロップダウン リストをクリックします。後続の検索ページには、選択した検索タイプに固有の画面コンポーネントが含まれます。

Manage Devices ページでは、別々であっても関連のある 2 つの領域を使用して検索結果が生成され、これによりネットワーク内のデバイスを管理できます。表示される領域は次の 2 つです。

- Search Type ドロップダウン リスト。実行する検索を定義します。
- 追加の値フィールド。選択した検索タイプを指定します。これらのフィールドには、IP Address、MAC Address または MAC Address ワイルドカード、Node Name（Node Type）、および Owner ID などがあります。

Manage Devices ページから、次の検索を実行できます。

- DUID Search：IPv6 環境でデバイスの DHCP Unique Identifier（DUID）を使用して検索します。DUID で許容されるのは、ネットワーク バイト オーダーで表された 2 オクテット タイプ コードが先頭にあり、その後に ID を形成するさまざまな数のオクテットが続く形式です。たとえば、00:03:00:01:02:03:04:05:07:a0 のようになります。この検索基準を有効に使用する方法の詳細については、P.16-3 の「デバイス ID に基づくデバイスのトラブルシューティング」を参照してください。
- FQDN Search：DNS サーバによって割り当てられるデバイスに関連付けられている Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を使用して検索します。この検索は、デバイスの MAC アドレスが不明な場合に特に有効です。たとえば、**www.myhost.example.com** は完全修飾ドメイン名です。**myhost** はホストを、**example** は第 2 レベル ドメインを、また **.com** は第 3 レベル ドメインを指定します。

- IP Address Search：現在指定された DHCP リース IP アドレスを持つ、ネットワーク上のすべてのデバイスを返すことによって検索します。
- MAC Address Search：正確な MAC アドレスを使用して、特定のモデム、または装置ベンダーを明確に識別する特定のベンダープレフィックスを持つすべてのデバイスを検索します。ベンダープレフィックスは、MAC アドレスの先頭の 3 オクテットです。たとえば、MAC アドレスが 1,6,aa:bb:cc:dd:ee:ff の場合、ベンダープレフィックスは「aa:bb:cc」です。したがって、MAC Address Search を実行する場合は、デバイスの製造業者およびタイプを指定できます。この検索基準を有効に使用する方法の詳細については、P.16-3 の「デバイス ID に基づくデバイスのトラブルシューティング」を参照してください。
- Node Search：特定のノードまたはノード タイプの一部になっているデバイスを検索します。
- Owner ID Search：デバイスに関連付けられているオーナー ID を使用して検索します。オーナー ID は、サービス加入者のアカウント番号などを示す場合があります。この検索機能では、ワイルドカード検索はサポートされていません。
- Provisioning Group Search：デバイスが属しているプロビジョニング グループを使用して検索します。
- Class of Service Search には、次の検索があります。
  - Registered Class of Service Search：デバイスにプロビジョニングされているサービス クラスを使用して検索します。
  - Related Class of Service Search：登録されているサービス クラスと選択されているサービス クラスの両方を使用して検索します。
  - Selected Class of Service Search：RDU によって選択されたサービス クラスを使用して、登録されているサービス クラスを何らかの理由で保持できないデバイスを検索します。
- DHCP Criteria Search には、次の検索があります。
  - Registered DHCP Criteria Search：特定の DHCP 基準に属するデバイスを検索します。
  - Related DHCP Criteria Search：登録されている DHCP 基準と選択した DHCP 基準の両方を使用して検索します。
  - Selected DHCP Criteria Search：RDU によって選択された DHCP 基準を使用して、登録されている DHCP 基準を何らかの理由で保持できないデバイスを検索します。

（注） 通常、Related Class of Service と Selected Class of Service、Related DHCP Criteria と Selected DHCP Criteria は同一です。同一でない場合は、Selected Class of Service または Selected DHCP Criteria を調査し、Related Class of Service または Related DHCP Criteria と一致するように修正してください。

実行可能な検索の中には、ワイルドカード文字（*）を使用して検索機能を拡張できるものがあります。BAC には、各検索で使用できる特定のワイルドカードがあります。表 12-1 は、それらのワイルドカードを示しています。

（注） 数十万単位のデバイスをサポートするシステムでワイルドカード検索（*）を使用することは推奨されません。そのような検索を行うと、結果が数千の規模になり、システム リソースの使用量が増大してパフォーマンスに悪影響を与えます。

表 12-1　デバイス管理でサポートされる検索

検索メニュー	検索タイプのオプション
DUID Search	DUID 全体、または文字列の最後にワイルドカード文字（*）が続く部分的な DUID。 たとえば、DUID が 00:03:00:01:02:03:04:05:06:a0 のデバイスを検索するには、00:03:* と指定できます。
FQDN Search	FQDN 全体、または先頭にワイルドカード文字（*）を使用した部分的な FQDN 文字列。 たとえば、FQDN が IGW-1234.EXAMPLE.COM のデバイスを検索するには、次のように指定できます。 • *.example.com • *.com • *
IP Address Search	IP Address ワイルドカード検索はサポートされていません。完全な IP アドレスを入力する必要があります。 たとえば、IP アドレスが 10.10.10.10 のデバイスを検索するには、10.10.10.10 と入力する必要があります。
MAC Address Search	MAC アドレス全体、または文字列の最後にワイルドカード文字（*）が続く部分的な MAC アドレス。 たとえば、MAC アドレスが 1,6,aa:bb:cc:dd:ee:ff のデバイスを検索するには、1,6,* と指定できます。
Node Search	Node Name (Node Type)：ドロップダウン リストにあるノードおよびノード タイプ。オプションとして、デフォルトの **system-diagnostics (system)** オプションや、定義したその他のノードを指定できます。
Owner ID Search	Owner ID ワイルドカード検索はサポートされていません。完全なオーナー ID を入力する必要があります。 たとえば、オーナー ID が 100000000000xxxxx のデバイスを検索するには、100000000000xxxxx と入力する必要があります。
Provisioning Group Search	Provisioning Group name ドロップダウン リストから、**デフォルト** オプションを選択するか、その他の自分で設定したプロビジョニング グループを選択します。
Registered Class of Service Search	Class of Service (Type) ドロップダウン リストから、デフォルト オプションを選択するか、Registered Class of Service として定義したその他のオプションを選択します。
Registered DHCP Criteria Search	DHCP Criteria (Type) ドロップダウン リストから、デフォルト オプションを選択するか、Registered DHCP Criteria として定義したその他のオプションを選択します。

表 12-1　デバイス管理でサポートされる検索（続き）

検索メニュー	検索タイプのオプション
Related Class of Service Search	Class of Service (Type) ドロップダウン リストから、デフォルト オプションを選択するか、Related Class of Service として定義したその他のオプションを選択します。
Related DHCP Criteria Search	DHCP Criteria (Type) ドロップダウン リストから、デフォルト オプションを選択するか、Related DHCP Criteria として定義したその他のオプションを選択します。
Selected Class of Service Search	Class of Service (Type) ドロップダウン リストから、デフォルト オプションを選択するか、Selected Class of Service として定義したその他のオプションを選択します。
Selected DHCP Criteria Search	DHCP Criteria (Type) ドロップダウン リストから、デフォルト オプションを選択するか、Selected DHCP Criteria として定義したその他のオプションを選択します。

図 12-1 に、DUID 検索オプションを使用してデバイスを検索するときの Manage Devices ページのサンプルを示します。

図 12-1　Manage Devices ページ

Manage Devices ページの Page Size ドロップダウン リストでは、表示されるページあたりの検索結果数を制限できます。結果表示件数は、25、50、または 75 から選択できます。検索で返される結果の数が選択した数を超えると、ページの左下に画面プロンプトが表示されます。これらのコントロールを使用して、1 ページずつ前後にスクロールしたり、特定のページを選択したりできます。

任意のクエリーから返される結果の最大数は 1,000 で、1 ページに表示される結果の最大数は 75 です。デフォルトの最大値を変更するには、次の手順に従います。

1. *BPR_HOME/rdu/conf/adminui.properties* ファイルの */adminui/maxReturned* プロパティを変更します。
2. 管理者ユーザ インターフェイスの BAC Tomcat プロセスを再起動します。

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

デバイス管理コントロール

デバイス管理コントロールは、検索機能フィールドのすぐ下に配置されており、通常は検索機能で使用します。たとえば、特定のデバイス グループに属するデバイスを検索して、何らかの管理機能を実行することができます。

次のボタンを使用できますが、使用する検索タイプによってはそれぞれの管理機能を利用できない場合があります。

- **Add**：Add ボタンを使用して、新しいデバイスを RDU データベースに追加します。P.12-14 の「デバイス レコードの追加」を参照してください。
- **Delete**：Delete ボタンを使用して、RDU データベースからデバイスを削除します。P.12-15 の「デバイスの削除」を参照してください。
- **Regenerate**：Regenerate ボタンを使用して、選択したデバイスの構成の再生成を強制的にすぐ実行させます。P.12-16 の「デバイス構成の再生成」を参照してください。
- **Relate**：Relate ボタンを使用して、デバイス（MAC アドレスまたは DUID を使用）を特定のノードと関連付けます。P.12-17 の「デバイスの関連付けと関連付け解除」を参照してください。
- **Reset**：Reset ボタンを使用して、自動的にデバイスをリブートします。
- **Unrelate**：Unrelate ボタンを使用して、選択したデバイスと、そのデバイスが現在関連付けられているノードとの関連付けを解除します。P.12-17 の「デバイスの関連付けと関連付け解除」を参照してください。

デバイスを検索すると、次に示すヘッダーまたはリンクがページに表示され、その下に結果が返されます。

- Identifier：検索基準と一致するすべてのデバイスを示します。表示される各 ID には、そのデバイスを修正できる別ページへのリンクが設定されます。
- Device Type：利用可能なデバイス タイプを表示します。選択肢として、次のデバイス タイプがあります。
  - CableHome MAN-Data
  - CableHome MAN-WAN
  - DOCSIS Modem
  - Computer
  - PacketCable Multimedia Terminal Adapter (MTA)
  - Set-top box (STB)
- Status：デバイスがプロビジョニングされるかどうかを示します。プロビジョニングされるデバイスは、Application Programming Interface（API; アプリケーション プログラミング インターフェイス）、つまり管理者ユーザ インターフェイスを使用して登録されており、ネットワークでブートされているデバイスです。
- Details：選択したデバイスに関する利用可能なすべての詳細情報が表示されます。詳細については、P.12-10 の「デバイスの詳細の表示」を参照してください。

デバイスの詳細の表示

検索結果で示された任意のデバイスの詳細を表示できます。任意のデバイスの詳細を表示するには、表示するデバイスの **View Details** アイコン（）をクリックして View Device Details ページを表示します。

（注） View Device Details ページに表示される情報は、選択するデバイスのタイプに応じて決まります。表 12-2 で使用されるサンプル図は、ほとんどのデバイスの場合に標準的に表示される詳細を示しています。

表 12-2　View Device Details ページ

フィールドまたはボタン	説明
Device Details	
Device Type	たとえば DOCSIS モデムなどのデバイス タイプを示します。
MAC Address	デバイスの MAC アドレスを示します。
DUID	デバイスの DUID を示します。
FQDN	デバイスの完全修飾ドメイン名（FQDN）を示します（IGW-1234.EXAMPLE.COM など）。
Host Name	ホストを示します。たとえば、上記の FQDN の場合、IGW-1234 がホスト名です。
Domain Name	ホストが存在するドメインを示します。たとえば、上記の FQDN の場合は EXAMPLE.COM がドメイン名です。
OID	MIB データベースの特定の SNMP オブジェクトを示す値である、オブジェクト識別子を示します。
Revision Number	処理前に検証される OID リビジョン番号を示します。
Behind Device	このデバイスの背後にあるデバイスを示します。
Provisioning Group	デバイスが事前に、または自動的に割り当てられているプロビジョニング グループを示します。これはアクティブ リンクで、クリックすると、Provisioning Group Details ページが表示されます。
Registered DHCP Criteria	使用される DHCP 基準を示します。デフォルトの DHCP 基準の場合を除いてアクティブ リンクとなり、クリックすると、該当する Modify DHCP Criteria ページが表示されます。デフォルトの DHCP 基準を選択すると、Systems Defaults ページでデフォルトとして設定されている DHCP 基準が適用されます。
Device Properties	このデバイスに設定できる、このページに表示されていないプロパティを示します。このフィールドには、カスタム プロパティの表示が含まれます。
Device Provisioned State	デバイスがプロビジョニングされるかどうかを示します。デバイスがプロビジョニングされるのは、登録されており、ネットワーク上でブートされている場合のみです。
Device Registered State	デバイスが登録されているかどうかを示します。
Client Identifier	DHCP メッセージでデバイスが使用するクライアント ID を示します。
Client Request Host Name	クライアントがその DHCP メッセージの中で要求するホスト名を示します。

表 12-2　View Device Details ページ （続き）

フィールドまたはボタン	説明
Registered Class of Service	デバイスに割り当てられたサービス クラスを示します。これはアクティブ リンクで、クリックすると、該当する Modify Class of Service ページが表示されます。 拡張により、デバイスで別のサービス クラスが選択されている場合は、Selected Class of Service という追加フィールドが表示されます。
Owner Identifier	デバイスを示します。これは、ユーザ ID またはアカウント番号です。このフィールドは空白にすることもできます。
Detected Properties	デバイスの構成が生成されるときに、RDU デバイス検出拡張によって返されるプロパティを示します。
Selected Properties	デバイスの構成が生成されるときに、検出されたデバイス タイプの RDU サービス レベル選択拡張によって返されるプロパティを示します。
Is Behind Required Device	必要なリレー エージェント デバイスを確立するために *DeviceDetailsKeys.IS_BEHIND_REQUIRED_DEVICE* プロパティが使用されており、サービス レベル選択拡張により、必要なリレー エージェントの背後でこのデバイスがブートしなかったと判断される場合は、「false」と表示されます。
Is In Required Provisioning Group	必要なプロビジョニング グループを確立するために *IPDeviceKeys.MUST_BE_IN_PROV_GROUP* プロパティが使用されており、サービス レベル選択拡張により、必要なプロビジョニング グループでこのデバイスがブートしなかったと判断される場合は、「false」と表示されます。
Selected Access	サービス レベル選択拡張によってデバイスに付与されるアクセス権を示します。 • REGISTERED：デバイスが登録されてアクセス要件が満たされたことを示します。 • PROMISCUOUS：デバイスのリレー エージェントに割り当てられるポリシーが、そのデバイスのプロビジョニングの基盤となることを示します。 • DEFAULT：デバイスが、そのデバイス　タイプのデフォルト アクセス権によってプロビジョニングされることを示します。 • OTHER：BAC　に組み込まれるデフォルト拡張では使用されず、カスタム拡張で使用するために用意されています。
Selected Class of Service	デバイスの構成を生成するために使用されるサービス クラスの名前を示します。これはアクティブ リンクで、クリックすると、該当する Modify Class of Service ページが表示されます。
Selected DHCP Criteria	デバイスの構成を生成するために使用される DHCP 基準の名前を示します。これはアクティブ リンクで、クリックすると、該当する Modify DHCP Criteria ページが表示されます。
Selected Explanation	そのデバイスに付与したアクセス権がサービス レベル選択拡張で選択された理由についての説明を示します。たとえば、必要なプロビジョニング グループでブートしなかったために、そのデバイスにデフォルトのアクセス権が与えられている場合があります。

表 12-2　View Device Details ページ （続き）

フィールドまたはボタン	説明
Selected Reason	そのデバイスに与えたアクセス権がサービス レベル選択拡張で選択された理由を列挙コードとして示します。次の値があります。 • NOT_BEHIND_REQUIRED_DEVICE • NOT_IN_REQUIRED_PROV_GROUP • NOT_REGISTERED • OTHER • PROMISCUOUS_ACCESS_ENABLED • REGISTERED • RELAY_NOT_IN_REQUIRED_PROV_GROUP • RELAY_NOT_REGISTERED これらの値のほとんどは、登録されたアクセス権または無差別のアクセス権を与える際に要件違反があり、結果としてデフォルトのアクセス権が付与されていることを示します。
Related Node Name (Node Type)	このデバイスが関連付けられているノードを示します。これはアクティブ リンクで、クリックすると、該当する Modify Node ページが表示されます。P.12-19 の「ノード管理」を参照してください。
DHCPv4 Information	
（注） このセクションは、デバイスが DHCPv4 データを検出しなければ表示されません。	
DHCP Inform Dictionary	構成の生成を要求するときに、Cisco Network Registrar 拡張が RDU に送信する追加情報を示します。これは内部 BAC でのみ使用されます。
DHCP Request Dictionary	構成の生成を要求するときに、Network Registrar 拡張から RDU に送信される DHCP 検出パケットまたは DHCP 要求パケットの詳細情報を示します。
DHCP Response Dictionary	このフィールドは内部 BAC でのみ使用され、常に空の状態です。
DHCP Environment Dictionary	このフィールドは内部 BAC でのみ使用され、常に空の状態です。
Lease v4 Information	
（注） このセクションは、デバイスが Lease v4 データを検出しなければ表示されません。	
IP Address	デバイスの IPv4 アドレスを示します。
DHCP Lease Properties	IPv4 アップデートと一緒に、Network Registrar が RDU に送信するリース プロパティを示します。

表 12-2　View Device Details ページ （続き）

フィールドまたはボタン	説明
DHCPv6 Information	
（注） このセクションは、デバイスが DHCPv6 データを検出しなければ表示されません。	
DHCPv6 Inform Dictionary	構成の生成を要求するときに、Cisco Network Registrar 拡張が RDU に送信する追加情報を示します。これは内部 BAC でのみ使用されます。
DHCPv6 Request Dictionary	構成の生成を要求するときに、Network Registrar 拡張から RDU に送信される DHCP 検出パケットまたは DHCP 要求パケットの詳細情報を示します。
DHCPv6 Relay Request Dictionary	構成の生成を要求するときに、Network Registrar 拡張から RDU に送信される DHCP パケットの詳細情報を示します。ただし、このデータは CMTS から派生し、CMTS についての情報と、CMTS が使用する DOCSIS バージョンの情報が含まれています。
DHCPv6 Response Dictionary	このフィールドは内部 BAC でのみ使用され、常に空の状態です。
DHCPv6 Environment Dictionary	このフィールドは内部 BAC でのみ使用され、常に空の状態です。ただし、Network Registrar Default（**Configuration** > **Defaults** > **NR Defaults**）ページの Environment Dictionary から Attributes の値を設定する場合は、ここにその値が表示されます。
Lease v6 Information	
（注） このセクションは、デバイスが Lease v6 データを検出しなければ表示されません。	
IP Address	デバイスの IPv6 アドレスを示します。
DHCPv6 Lease Properties	IPv6 アップデートと一緒に、Network Registrar が RDU に送信するリース プロパティを示します。
Technology-Specific Information	
（注） テクノロジー固有情報には、使用ライセンスを持っているテクノロジーに関連するデータのみが示されます。	
XGCP Ports	Gateway Control Protocol がアクティブになっているポートを示します。
DOCSIS のバージョン	現在使用中の DOCSIS バージョンを示します。

デバイスの管理

Devices メニューを使用すると、RDU データベースにデバイスを追加し、プロビジョニングされたデータを更新することができます。デバイス管理には、次の作業があります。

- RDU デバイス レコードの追加、削除、および修正。
- 構成の再生成。
- 管理オブジェクト（プロビジョニング グループ、サービス クラス、グループなど）へのデバイスの関連付け。

この項では、新しいデバイスまたは既存のデバイスに対して、各種のデバイス管理機能を実行する方法について説明します。いくつかの情報フィールドがデバイス管理ページすべてに一貫して表示されます。次のフィールドがあります。

- Device Type：デバイスを追加する場合にドロップダウン リストとして表示され、BAC 内で作成することができる利用可能なデバイス タイプを示します。画面に表示される選択肢には次のものがあります。
  - CableHomeWanData
  - CableHomeWanMan
  - Computer
  - DOCSISModem
  - PacketCableMTA
  - STB

  デバイスを修正するときにデバイス タイプを編集または変更することはできません。
- MAC Address：デバイスの MAC アドレスを示します。

  追加するデバイスの MAC アドレスをこのフィールドに入力します。入力するときには、カンマ（,）とコロン（:）も正しく入力してください。たとえば、1,6,00:00:00:00:00:00:AE のように入力します。
- DUID：デバイスの DUID を示します。

  追加するデバイスの DUID をこのフィールドに入力します。入力するときには、コロン（:）も正しく入力してください。たとえば、00:03:00:01:02:03:04:05:06:a0 のように入力します。
- Host Name：デバイス ホストを示します。たとえば、node.example.com という FQDN の場合は、node がホスト名です。
- Domain Name：ホストが存在するドメインを示します。たとえば、node.example.com という FQDN の場合は、example.com がドメイン名です。
- Owner Identifier：ホスト名以外の情報を使用してデバイスを示します。たとえば、ユーザ ID や 1000000000000000000 のようなアカウント番号が表示されます。このフィールドは空白のままにすることもできます。
- Registered Class of Service：デバイスがプロビジョニングされるサービス クラスを指定します。たとえば、デフォルト オプションや自分で定義したサービス クラスです。
- Registered DHCP Criteria：デバイスがプロビジョニングされる DHCP 基準を指定します。たとえば、デフォルト オプションや自分で定義した DHCP 基準です。

デバイス レコードの追加

デバイス レコードを追加するには、次の手順に従います。

ステップ 1　Manage Devices ページから **Add** をクリックします。

Add Device ページが表示されます。

ステップ 2　ドロップダウン リストに表示される選択肢からデバイス タイプを選択します。

ステップ 3　ページの他のフィールドに、MAC アドレス、DUID、およびホスト名などの詳細情報を入力します。

ステップ 4　サービス クラス、およびそのデバイスについて登録する DHCP 基準を選択します。

ステップ 5　デバイスについてここまでで入力した値に加えて、オプションで、既存のプロパティ名と値のペアに新しい値を追加できます。

- Property Name：カスタムまたは組み込みデバイス プロパティの名前を示します。
- Property Value：プロパティの値を示します。

ステップ 6　**Submit** をクリックします。

デバイス レコードの修正

デバイス レコードを修正するには、次の手順に従います。

ステップ 1　Manage Devices ページから、デバイスに対応する Identifier リンクをクリックします。

Modify Device ページが表示されます。

ステップ 2　追加または変更するデータを入力します。**Add** をクリックして既存の任意のプロパティ名と値のペアを修正するか、**Delete** をクリックして任意のペアを削除します。

ステップ 3　**Submit** をクリックして、このデバイスへの変更を保存します。

入力した値を削除するには、**Reset** をクリックします。

デバイスの削除

デバイス レコードの削除は単純な手順ですが、慎重に使用する必要があります。削除を取り消すには、以前バックアップしたデータベースを復元するか、そのデバイスを再度追加する必要があります。バックアップしたデータベースの復元が必要になった場合は、P.15-7 の「データベースの復元」を参照してください。

デバイス レコードを削除するには、次の手順に従います。

ステップ 1　Manage Devices ページで、削除するデバイスを検索します。そのためには、いずれかの検索タイプを使用します。

ステップ 2　デバイスの左にあるチェックボックスをオンにします。

ステップ 3　**Delete** をクリックします。

RDU データベースにストアされているデバイス レコードが削除されます。

デバイス構成の再生成

Regenerate ボタンまたは API 操作を使用すると、デバイスの構成の再生成を即時に実行できます。デバイスの構成は、デバイスのプロビジョニング グループにある DPE に送信されます。

通常、構成を再生成するプロセスは、デバイスやサービス クラスに対する変更、または影響を及ぼすその他の変更の後に自動的にトリガーされます。ただし、サービス クラスに対する変更が行われた後は、システムがすべてのデバイスに対する構成を再生成するまで時間がかかります。Regenerate ボタンを使用して、指定されたデバイスの構成の再生成を迅速に実行できます。このオプションは、予防的なトラブルシューティングを行うときに特に便利です。

場合によっては、サービス クラスまたは DHCP 基準のパラメータの多くを変更することが必要になります。その場合は、既存のデバイス構成が古くなり、構成を再生成することが必要になります。各構成を手動で再生成する必要性をなくし、エラーが紛れ込む可能性を低くするため、BAC には、すべてのデバイス構成を自動的に再生成するために使用できる Configuration Regeneration Service（CRS; 構成再生成サービス）が用意されています。

デバイス構成は、次の場合に自動的に再生成されます。

- サービス クラスに関連付けられたファイル（つまりテンプレート）が更新されたとき。
- デバイス タイプのデフォルトのサービス クラスまたは DHCP 基準が変更されたとき。
- DHCP 基準プロパティが変更されたとき。
- プロビジョニング グループ オブジェクトが管理者のユーザ インターフェイスまたは API を介して変更されたとき。
- サービス クラス オブジェクトのプロパティが変更されたとき。
- DPE が構成再生成要求を RDU に送信したとき。
- デバイスのプロパティまたは関連付けが更新されたとき。

加えられた変更がデバイス構成に影響するかどうかを BAC は判別できないので、一部の構成は自動的には再生成できません。そのような場合は、*generationConfiguration()* メソッドまたは管理者ユーザ インターフェイスを使用して、構成を手動で再生成してください。手動で構成を再生成する必要があるのは、次の場合です。

- デフォルトのサービス クラスとデフォルトの DHCP 基準の場合を除いて、テクノロジー デフォルトが変更されたとき。デフォルトのサービス クラスと DHCP 基準のテクノロジー デフォルト プロパティを変更しても、デフォルトの DHCP 基準またはデフォルトのサービス クラスが指定されているデバイスの再生成はトリガーされません。
- システム デフォルトが変更されたとき。
- 別の DOCSIS テンプレート内に含まれているファイルが変更されたとき。

（注） 構成が再生成される方法に関係なく、デバイス構成が有効になるまで構成はデバイスに伝播されません。デバイス構成が有効になるのは、デバイスがスケジュールに従って DPE に接続するか、DPE から開始された接続要求の結果として DPE に接続したときです。

デバイスの構成を再生成するには、次の手順に従います。

ステップ 1　Manage Devices ページで、構成を再生成するデバイスを検索します。そのためには、いずれかの検索タイプを使用します。

ステップ 2　デバイスの左にあるチェックボックスをオンにします。

ステップ 3　**Regenerate** をクリックします。

RDU は、特定のデバイスの構成を再生成します。

デバイスの関連付けと関連付け解除

デバイスの関連付けという概念は、デバイスが特定のサービス クラスまたは特定の DHCP 基準に関連する限り、サービス クラスまたは DHCP 基準の関連付けと類似しています。大きな違いは、サービス クラスと DHCP 基準は事前定義ノードと見なされることと、ノードを使用して、定義する任意グループにデバイスをグループ化することです。

このコンテキストでは、関連付け機能により、MAC アドレスまたは DUID を使用してデバイスを特定のノードに関連付けることができます。さらにその特定のノードは、特定のノード タイプに関連付けられます。

デバイスを特定のノードに関連付けることにより、デバイスが特定のノードに関連付けられていることを示す情報がデータベースに保存されます。事前定義された **system-diagnostics (system)** ノードにデバイスを関連付ける場合は、利用可能な情報を使用して潜在的な問題のトラブルシューティングを行うことができます。

ノードへのデバイスの関連付け

管理者ユーザ インターフェイスから関連付けまたは関連付け解除できるデバイスは、一度に 1 つだけです。

デバイスを関連付けるには、次の手順に従います。

ステップ 1　Manage Devices ページで、ノードに関連付けるデバイスを検索します。そのためには、いずれかの検索タイプを使用します。

ステップ 2　デバイスの左にあるチェックボックスをオンにします。

ステップ 3　**Relate** をクリックします。

Relate Device to Node ページが表示されます。

ステップ 4　ドロップダウン リストからノード タイプを、定義済みノードのリストからノードを選択します。

（注）　Nodes リストから複数のグループを選択するには、**Control** キーまたは **Shift** キーを押します。

ステップ 5　**Submit** をクリックします。

指定したノードにデバイスが関連付けられていることを確認するには、そのデバイスに対応する View Details アイコンをクリックします。表示された Device Details ページで、Related Node Name（Node Type）の状態を確認します。

ノードからのデバイスの関連付け解除

デバイスをノードから関連付けを解除するには、次の手順に従います。

ステップ 1　Manage Devices ページで、ノードから関連付け解除するデバイスを検索します。

ステップ 2　デバイス ID に対応するチェックボックスをオンにし、**Unrelate** ボタンをクリックします。

Unrelate Device from Node ページが表示されます。

ステップ 3　定義済みノードのリストから、デバイスを関連付け解除するノードを選択します。

（注）　Nodes リストから複数のグループを選択するには、**Control** キーまたは **Shift** キーを押します。

ステップ 4　**Submit** をクリックします。

Manage Devices ページが表示されます。

ノードでのデバイスの検索

特定のノードに属するデバイスを検索するには、次の手順に従います。

ステップ 1　Manage Devices ページで、Search Type の下のドロップダウン リストから Node Search オプションを選択します。

Node Name（Node Type）が表示されます。

ステップ 2　Node Name（Node Type）ドロップダウン リストから、デバイスが関連付けられているノードの名前を選択します。

ステップ 3　**Search** をクリックします。

ノードに関連付けられているデバイスが表示されます。

デバイスのリセット

Reset ボタンを使用して選択したデバイスをリブートできます。

デバイスをリセットするには、次の手順に従います。

ステップ 1　Manage Devices ページで、リブートするデバイスを検索します。そのためには、いずれかの検索タイプを使用します。

ステップ 2　そのデバイスに対応するチェックボックスをオンにします。

ステップ 3　**Reset** をクリックします。

デバイスがリブートします。

ノード管理

ノード管理により、ノードおよびノード タイプを作成、変更、削除できます。BAC のコンテキスト内では、ノード自体がノード タイプを構成する一方で、ノード タイプをノードのグループと見なすことができます。

ノード タイプの管理

Main Menu またはプライマリ ナビゲーション バーから Nodes を選択して、Manage Nodes ページを開きます。このページを表示すると、デフォルト設定では Node Type が選択されています。

ノード タイプの追加

新しいノード タイプを追加するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、**Add** をクリックします。

Add Node Type ページが表示されます。

ステップ 2　新しいノード タイプの名前を入力します。

（注）　前の手順でカスタム プロパティを追加している場合は、ドロップダウン リストから適切な Property Name を選択し、必要な Property Value を入力します。**Add** をクリックして、該当する Property Name と Property Value のペアの数を増やします。

新しいノード タイプが表示されます。

ステップ 3　**Submit** をクリックします。

新しいノード タイプが RDU に記録され、その新しいノード タイプが追加された状態で Manage Node ページが表示されます。

ノード タイプの修正

ノード タイプのプロパティを修正するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、特定のノード タイプをクリックします。

Modify Node Type ページが表示されます。

（注） 前の手順でカスタム プロパティを追加している場合は、Property Name/Property Value ペアに必要な変更を加えることができます。特定のペアを削除する必要がある場合は、そのペアの隣にある **Delete** をクリックします。

ステップ 2　**Submit** をクリックします。

情報が変更された状態で Manage Node ページが表示されます。

ノード タイプの削除

ノード タイプを削除するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、削除するノード タイプに対応する **Delete** アイコン（）をクリックします。

ステップ 2　表示される確認ダイアログボックスで **OK** をクリックし、選択したノード タイプを削除します。

削除したノード タイプが消えた状態で Manage Nodes ページが表示されます。

ノードの管理

ノードの作成と修正、不要なノードの削除、ノードとノード タイプの関連付けと関連付け解除、およびノードに関連付けたデバイスの表示を行うことができます。

新規ノードの追加

新規ノードを追加するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、Search Type ドロップダウン リストから **Nodes** を選択します。

ステップ 2　**Add** をクリックします。

Add Node ページが表示されます。

ステップ 3　新規ノードの名前を入力して、このノードに使用する適切な Node Type を選択します。

（注） 前の手順でカスタム プロパティを追加している場合は、ドロップダウン リストから適切な Property Name を選択し、必要な Property Value を入力します。**Add** をクリックして、該当する Property Name と Property Value のペアの数を増やします。

ステップ 4　**Submit** をクリックします。

新規ノードが RDU に記録され、その新しいノードが追加された状態で Manage Nodes ページが表示されます。

ノードでのデバイスの検索

ノードに関連付けられたデバイスを表示するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、Search Type ドロップダウン リストから Nodes オプションを選択します。

ステップ 2　ノード タイプまたはノード名を基準として選択して検索できます。

- By Node Type：事前定義ノードのドロップダウン リストが表示されます。
- By Node Name：Node or Node Wildcard フィールドが表示されます。このフィールドには、ノードの名前またはワイルドカード文字（*）を入力できます。

ステップ 3　**Search** をクリックします。

ステップ 4　ノードに対応する Devices パラメータの下にある **View Details** アイコンをクリックします。

Manage Devices ページに Node Search 機能が表示されます。

ステップ 5　Manage Devices ページで、該当するノード タイプを選択します。検索機能の詳細については、P.12-5 の「デバイスの検索」を参照してください。

ノードに関連付けられているデバイスが表示されます。

ノードの修正

ノードのプロパティを修正するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、適切なノード リンクをクリックします。

Modify Node ページが表示されます。

（注）　前の手順でカスタム プロパティを追加している場合は、Property Name/Property Value ペアに必要な変更を加えることができます。特定のペアを削除する必要がある場合は、そのペアの隣にある **Delete** をクリックします。

ステップ 2　**Submit** をクリックします。

説明が変更された状態で Manage Nodes ページが表示されます。

ノードの削除

Manage Nodes ページに表示されるノードを削除するには、そのノードに対応するチェックボックスをオンにして、**Delete** をクリックします。

ノードがデータベースから削除されます。

ノード タイプからノードへの関連付け / 関連付け解除

関連付け機能と関連付け解除機能は、特定のノードとノード タイプの間の関連性を確立するために使用します。

この関連性を確立または解除するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、Search Type ドロップダウン リストから Nodes を選択します。

ステップ 2　ノード　タイプまたはノード名の検索基準を使用して、ノードと関連付けまたは関連付け解除するノード タイプを選択します。

ステップ 3　**Search** をクリックします。

指定したノードが表示されます。

ステップ 4　Relate to Node リンクまたは Unrelate from Node リンクをクリックします。

選択したリンクに応じて、Relate Node ページまたは Unrelate Node ページが表示されます。

ステップ 5　ドロップダウン リストから適切な Node Type を選択し、ノードを関連付けまたは関連付け解除する対象のノードを選択します。

ステップ 6　**Submit** をクリックします。

Manage Nodes ページが表示されます。

ノードの詳細の表示

ノードに関連した詳細を表示するには、次の手順に従います。

ステップ 1　Manage Nodes ページで、Search Type ドロップダウン リストから Nodes オプションを選択します。

ステップ 2　ノード タイプまたはノード名の検索基準を使用して、詳細を表示するノードを選択します。

ステップ 3　**Search** をクリックします。

ステップ 4　詳細を表示するノードに対応するリンクをクリックします。

Modify Node ページが表示され、ノード名とノード タイプの詳細が示されます。

サーバの表示

この項では、BAC サーバ ページについて説明します。

Device Provisioning Engine の表示

Manage Device Provisioning Engines ページ（**Servers > DPEs**）で、現在 BAC データベースに登録されているすべての DPE のリストを監視できます。このページに表示される各 DPE 名は、その DPE の詳細を表示する別ページへのリンクになっています。DPE リンクをクリックして詳細ページを表示します。ページの表示内容は、表 12-3 で説明される詳細と類似しています。

（注） RDU は、DPE が RDU に接続するときに使用する DPE インターフェイスで DNS 逆ルックアップを実行することで、Network Registrar 拡張と DPE の名前を判別します。

表 12-3　View Device Provisioning Engines Details ページ

フィールドまたはボタン	説明
Device Provisioning Engine Details	
Host Name	DPE ホスト名を示します。
Port	DPE が RDU への接続を確立するときに使用した DPE ポート番号を示します。
IP Address	DPE の IP アドレスを示します。
Primary Provisioning Group(s)	選択した DPE が属するプライマリ プロビジョニング グループを示します。これはアクティブ リンクで、クリックすると、そのプロビジョニング グループの Provisioning Group Details ページが表示されます。
Secondary Provisioning Group(s)	選択した DPE が属するセカンダリ プロビジョニング グループを示します（この DPE がセカンダリ プロビジョニング グループに属している場合）。これはアクティブ リンクで、クリックすると、そのプロビジョニング グループの Provisioning Group Details ページが表示されます。
Properties	DPE で設定されているプロパティを示します。
Version	現在使用中の DPE ソフトウェアのバージョンを示します。
Up Time	DPE が最後に起動してから、動作が継続している合計期間を示します。

表 12-3　View Device Provisioning Engines Details ページ（続き）

フィールドまたはボタン	説明
State	DPE が動作可能かどうかを示します。示される状態には、次のものがあります。 • Registering • Initializing • Synchronizing • Ready • Offline 各状態の詳細については、P.2-8 の「DPE と RDU 間の同期」を参照してください。 （注）このフィールドに Offline と示されている場合、Uptime フィールド以降の詳細は表示されません。Offline 以外の状態にある DPE は、クライアント要求を処理する準備が整っています。
Protocol Services このセクションでは、DPE の TFTP プロトコルと ToD プロトコルの状態を示します。	
TFTPv4	TFTPv4 が DPE でイネーブルなのか、またはディセーブルなのかを示します。
TFTPv6	TFTPv6 が DPE でイネーブルなのか、またはディセーブルなのかを示します。
ToDv4	ToDv4 が DPE でイネーブルなのか、またはディセーブルなのかを示します。
ToDv6	ToDv6 が DPE でイネーブルなのか、またはディセーブルなのかを示します。
Registered Capabilities このセクションでは、RDU に登録しているこのプロビジョニング グループのすべての DPE の機能を示します。	
IPv4 - DOCSIS 1.0/1.1	この IPv4 モードの DPE で、DOCSIS 1.0 および 1.1 バージョンがイネーブルになっているかどうかを示します。
IPv4 - DOCSIS 2.0	この IPv4 モードの DPE で、DOCSIS 2.0 バージョンがイネーブルになっているかどうかを示します。
IPv4 - DOCSIS 3.0	この IPv4 モードの DPE で、DOCSIS 3.0 バージョンがイネーブルになっているかどうかを示します。
IPv4 - PacketCable	この IPv4 モードの DPE で、PacketCable 音声テクノロジーがイネーブルになっているかどうかを示します。
IPv4 - CableHome	この IPv4 モードの DPE で、ホーム ネットワーキング テクノロジーがイネーブルになっているかどうかを示します。
IPv6 - DOCSIS 3.0	この IPv6 モードの DPE で、DOCSIS 3.0 バージョンがイネーブルになっているかどうかを示します。

表 12-3　View Device Provisioning Engines Details ページ（続き）

フィールドまたはボタン	説明
Dynamic TFTP Compression	この DPE でダイナミック TFTP 圧縮がイネーブルになっているかどうかを示します。この機能をイネーブルにして、DPE に保存されるダイナミック設定のサイズを圧縮できます。ダイナミック TFTP 設定と併用する場合、この機能は、DPE キャッシュのサイズを大幅に縮小します。 **（注）** この機能は、プロビジョニング　グループのすべての DPE がサポートする場合にのみ、**Servers** > **Provisioning Groups** ページからイネーブルにすることができます。詳細については、P.2-17 の「プロビジョニング グループの機能」を参照してください。
Log File	
DPE Log File	View Details アイコンが表示されます。このアイコンをクリックすると、View Log File Contents ページが表示され、*dpe.log* の詳細を確認できます。
Cache Statistics	
Hits	DPE が最後に起動して以降に記録されたキャッシュ ヒットの数を示します。
Misses	DPE が最後に起動して以降に記録されたキャッシュ ミスの数を示します。
Lease Updates	更新された IPv4 リースおよび IPv6 リースの数を示します。
Files	DPE に現在保存されているキャッシュ ファイルの数を示します。
Configurations	キャッシュに保存されているデバイス設定ファイルの数を示します。
TFTP Statistics v4	
Packets Received	選択した DPE で受信した TFTPv4 パケットの数を示します。
Packets Dropped	DPE の過負荷のためにドロップされた TFTPv4 パケットの数を示します。
Packets Successful	正常に送信された TFTPv4 パケットの数を示します。
Packets Failed	送信中に損失した TFTPv4 パケットの数を示します。
TFTP Statistics v6	
Packets Received	選択した DPE で受信した TFTPv6 パケットの数を示します。
Packets Dropped	DPE の過負荷のためにドロップされた TFTPv6 パケットの数を示します。
Packets Successful	正常に送信された TFTPv6 パケットの数を示します。
Packets Failed	送信中に損失した TFTPv6 パケットの数を示します。
Time of Day Statistics v4	
Packets Received	選択した DPE で受信した Time of Day v4 パケットの数を示します。
Packets Dropped	DPE の過負荷のためにドロップされた Time of Day v4 パケットの数を示します。
Packets Successful	正常に送信された Time of Day v4 パケットの数を示します。
Packets Failed	送信中に損失した Time of Day v4 パケットの数を示します。

表 12-3　View Device Provisioning Engines Details ページ（続き）

フィールドまたはボタン	説明
Time of Day Statistics v6	
Packets Received	選択した DPE で受信した Time of Day v6 パケットの数を示します。
Packets Dropped	DPE の過負荷のためにドロップされた Time of Day v6 パケットの数を示します。
Packets Successful	正常に送信された Time of Day v6 パケットの数を示します。
Packets Failed	送信中に損失した Time of Day v6 パケットの数を示します。
PacketCable SNMP Statistics	
SNMP Informs Successful	正常に送信された通知要求の数を示します。
SNMP Sets Successful	成功した SNMP セットの数を示します。
SNMP Configuration Informs Successful	正常にプロビジョニングされたことを示す、PacketCable MTA から受信した SNMP 通知の数を示します。
SNMP Configuration Informs Failed	プロビジョニングに失敗したことを示す、PacketCable MTA から受信した SNMP 通知の数を示します。
PacketCable MTA Statistics	
MTA AP Requests Received	DPE が MTA から受信した AP-REQ メッセージの数を示します。
MTA AP Responses Sent	DPE が MTA に送信した AP-REP メッセージの数を示します。
PacketCable KDC Statistics	
KDC FQDN Requests Received	KDC が DPE に送信した FQDN-REQ メッセージの数を示します。
KDC FQDN Responses Sent	DPE が KDC に送信した FQDN-REP メッセージの数を示します。
Configured Network Interfaces	
Provisioning Group Communication	DPE が属するプロビジョニング グループに関連した詳細情報を示します。
IPv4 Provisioning	IPv4 プロビジョニング用に設定された DPE インターフェイスの詳細情報を示します。次の詳細が示されます。 • IPv4 アドレス • ポート番号 • FQDN **（注）** このセクションは、DPE インターフェイスが IPv4 プロビジョニング用に設定されている場合にのみ表示されます。
IPv6 Provisioning	IPv6 プロビジョニング用に設定された DPE インターフェイスの詳細情報を示します。次の詳細が示されます。 • IPv6 アドレス • ポート番号 • FQDN **（注）** このセクションは、DPE インターフェイスが IPv6 プロビジョニング用に設定されている場合にのみ表示されます。

Network Registrar 拡張ポイントの表示

Manage Network Registrar Extension Points ページ（**Servers > NRs**）には、RDU に登録され、BAC で使用するように設定されているすべての Network Registrar サーバの拡張ポイントがリストされます。Network Registrar サーバは、起動するときに RDU に自動的に登録されます。

このページに表示される各 Network Registrar 拡張ポイントは、その拡張ポイントについての詳細を表示する二次的なページへのリンクになっています。Network Registrar 拡張ポイント リンクをクリックして詳細ページを表示します。ページには、表 12-4 に示す詳細情報が表示されます。

表 12-4　Network Registrar Extension Point Details ページの表示

フィールドまたはボタン	説明
Network Registrar Extension Point Details	
Host Name	Network Registrar を実行しているシステムのホスト名を表示します。
IP Address	Network Registrar サーバの IP アドレスを示します。
Provisioning Group	Network Registrar サーバのプロビジョニング グループを示します。これはアクティブ リンクで、クリックすると、そのプロビジョニング グループの Provisioning Group Details ページが表示されます。
Properties	Network Registrar サーバに適用されるプロパティを示します。
Version	現在使用中の拡張ポイント ソフトウェアを示します。
Up Time	Network Registrar 拡張ポイントが最後に起動してから、動作が継続している合計時間を示します。この時間は、時、分、秒単位で示されます。
State	DPE が動作可能かどうかを示します。示される状態には、次のものがあります。 • Registering • Initializing • Synchronizing • Ready • Offline 各状態の詳細については、P.2-8 の「DPE と RDU 間の同期」を参照してください。 **（注）** このフィールドに Offline と示されている場合、Uptime フィールド以降のオプションは表示されません。Offline 以外の状態にある DPE は、クライアント要求を処理する準備が整っています。
Protocol Services	
DHCPv4	DHCPv4 がイネーブルであるか、ディセーブルであるかを示します。
DHCPv6	DHCPv6 がイネーブルであるか、ディセーブルであるかを示します。

表 12-4　Network Registrar Extension Point Details ページの表示 （続き）

フィールドまたはボタン	説明
Registered Capabilities	
IPv4 - DOCSIS 1.0/1.1	Network Registrar サーバに接続する DPE で、DOCSIS 1.0 および 1.1 バージョンが IPv4 モードでイネーブルになっているかどうかを示します。
IPv4 - DOCSIS 2.0	Network Registrar サーバに接続する DPE で、DOCSIS 2.0 バージョンが IPv4 モードでイネーブルになっているかどうかを示します。
IPv4 - DOCSIS 3.0	Network Registrar サーバに接続する DPE で、DOCSIS 3.0 バージョンが IPv4 モードでイネーブルになっているかどうかを示します。
IPv4 - PacketCable	Network Registrar サーバに接続する DPE で、PacketCable 音声テクノロジーが IPv4 モードでイネーブルになっているかどうかを示します。
IPv4 - CableHome	Network Registrar サーバに接続する DPE で、ホーム ネットワーキング テクノロジーが IPv4 モードでイネーブルになっているかどうかを示します。
IPv6 - DOCSIS 3.0	Network Registrar サーバに接続する DPE で、DOCSIS 3.0 バージョンが IPv6 モードでイネーブルになっているかどうかを示します。
Network Registrar Extension Point Statistics	
DHCPv4 Packets Received	受信した DHCPv4 パケットの数を示します。
DHCPv4 Packets Ignored	無視された DHCPv4 パケットの数を示します。
DHCPv4 Packets Dropped	ドロップされた DHCPv4 パケットの数を示します。
DHCPv4 Packets Successful	正常に転送した DHCPv4 パケットの数を示します。
DHCPv4 Packets Failed	転送に失敗した DHCPv4 パケットの数を示します。
DHCPv6 Packets Received	受信した DHCPv6 パケットの数を示します。
DHCPv6 Packets Ignored	無視された DHCPv6 パケットの数を示します。
DHCPv6 Packets Dropped	ドロップされた DHCPv6 パケットの数を示します。
DHCPv6 Packets Successful	正常に転送した DHCPv6 パケットの数を示します。
DHCPv6 Packets Failed	転送に失敗した DHCPv6 パケットの数を示します。
Device Provisioning Engine Details	

（注） 次のフィールドは、Network Registrar サーバに接続する DPE ごとに表示されます。

DPE	DPE の IP アドレスを示します。
Port	DPE が RDU への接続を確立するときに使用したポート番号を示します。
Type	この DPE がプライマリ DPE なのか、またはセカンダリ DPE なのかを示します。
Status	DPE が動作しているかどうかを示します。

プロビジョニング グループの表示

Manage Provisioning Groups ページ（**Servers > Provisioning Groups**）を使用して、現在のプロビジョニング グループをすべて監視できます。このリストに表示される各プロビジョニング グループは、そのグループの詳細ページへのリンクになっています。このリンクをクリックすると詳細ページが表示されます。詳細ページには、表 12-5 に示す詳細情報が表示されます。

表 12-5　View Provisioning Group Details ページ

フィールドまたはボタン	説明
Provisioning Group Details	
Name	Manage Provisioning Groups ページで選択したプロビジョニング グループ名を示します。
Primary Device Provisioning Engine	このプロビジョニング グループのプライマリ DPE のホスト名を示します。これはアクティブ リンクで、クリックすると、View Device Provisioning Engine Details ページが表示されます。
Secondary Device Provisioning Engine	このプロビジョニング グループのセカンダリ DPE のホスト名を示します。これはアクティブ リンクで、クリックすると、View Device Provisioning Engine Details ページが表示されます。
Network Registrar Extension Points	このプロビジョニング グループに割り当てられた Network Registrar サーバのホスト名を示します。これはアクティブ リンクで、クリックすると、View Network Registrar Extension Point Details ページが表示されます。
Number of Devices	このプロビジョニング グループに属するデバイスの数を示します。
Lease Query Management	
LeaseQuery AutoConfig	リース クエリー アドレスの自動設定をイネーブルまたはディセーブルにします。この機能は、デフォルトでイネーブルになっています。 この機能をイネーブルにすると、RDU は、そのリース クエリー設定を調整して、プロビジョニング グループの Network Registrar サーバから IPv4 アドレス リストと IPv6 アドレス リストの両方を設定します。 この機能をディセーブルにすると、RDU は、Network Registrar サーバに登録するときにリース クエリー設定を変更しません。 **（注）** この機能をディセーブルにする場合にのみ、以降のフィールドがこのセクションに表示されます。
Configured IP Address List (IPv4)	DHCPv4 リース クエリー要求を送信するときに使用するように RDU が設定されている Network Registrar 拡張の IPv4 アドレスのリストを表示します。
Configured IP Address List (IPv6)	DHCPv6 リース クエリー要求を送信するときに使用するように RDU が設定されている Network Registrar 拡張の IPv6 アドレスのリストを表示します。

表 12-5　View Provisioning Group Details ページ（続き）

フィールドまたはボタン	説明
Capabilities Management	
これらのフィールドを使用して、プロビジョニング グループの DPE がその機能に基づいて起動時に RDU に登録するデバイス タイプのサポートを、手動でイネーブルまたはディセーブルにします。フィールドが Disabled の場合は、プロビジョニング グループで指定されたデバイス タイプまたは機能をサポートすることができないことを意味します。P.2-17 の「プロビジョニング グループの機能」を参照してください。 これらのフィールドの値は次のとおりです。 • Enabled：サーバはイネーブルで、使用できるように設定されています。 • Disabled：サーバはその機能をサポートしていますが、使用できるように設定されていません。 • Not Capable：サーバはその機能をサポートしません。BAC 4.0 をアップグレードしてその機能のサポートをイネーブルにする必要があります。	
IPv4 - DOCSIS 1.0/1.1	DOCSIS 1.0 および 1.1 のモデム、およびそれらの背後にあるコンピュータの IPv4 モードでのサポートをイネーブルまたはディセーブルにします。この機能をサポートするには、プロビジョニング グループの DPE と、DHCPv4 をサポートする Network Registrar DHCP サーバで、TFTPv4 もイネーブルにする必要があります。
IPv4 - DOCSIS 2.0	IPv4 モードでのすべての DOCSIS 1.0 と 1.1 デバイスおよび DOCSIS 2.0 モデムのサポートをイネーブルまたはディセーブルにします。
IPv4 - DOCSIS 3.0	IPv4 モードでの DOCSIS 1.0、1.1、2.0、および 3.0 のモデム、およびこれらのモデムの背後にあるセットトップ ボックスのサポートをイネーブルまたはディセーブルにします。この機能をサポートするには、プロビジョニング グループ内のすべての DPE で BAC 4.0 が確実に実行されるようにします。
IPv4 - PacketCable	IPv4 モードでの PacketCable MTA のサポートをイネーブルまたはディセーブルにします。この機能をサポートするには、プロビジョニング グループのすべての DPE で PacketCable をイネーブルにする必要があります。
IPv4 - CableHome	IPv4 モードでのホーム ネットワーキング デバイスのサポートをイネーブルまたはディセーブルにします。
IPv6 - DOCSIS 3.0	Ipv6 モードでの DOCSIS 3.0 のモデム、およびこれらのモデムの背後にあるセットトップ ボックスのサポートをイネーブルまたはディセーブルにします。この機能をサポートするには、プロビジョニング グループの DPE と、DHCPv6 をサポートする Network Registrar DHCP サーバで、TFTPv6 をイネーブルにする必要があります。
Dynamic TFTP Compression	このプロビジョニング グループの DPE でのダイナミック TFTP 圧縮をイネーブルまたはディセーブルにします。この機能をイネーブルにすると、DPE がキャッシュするダイナミック TFTP ファイルが圧縮され、DPE のパフォーマンスが向上します。ネットワーク内のデバイスのほとんどで大きなサイズのファイルが使用される場合は、ダイナミック TFTP 圧縮をイネーブルにします。 この機能を使用するには、プロビジョニング グループ内のすべての DPE で BAC 4.0 が確実に実行されるようにします。

Regional Distribution Unit の詳細の表示

Server メニューの RDU オプションを使用して、表 12-6 に示される RDU の詳細情報を表示します。

表 12-6　View Regional Distribution Unit Details ページ

フィールドまたはボタン	説明
Regional Distribution Unit Details	
Host Name	RDU を実行しているシステムのホスト名を示します。
Port	DPE からの接続に使用する RDU リスニング ポート番号を示します。デフォルトのポート番号は 49187 ですが、RDU のインストール時に別のポート番号を選択できます。
IP Address	RDU に割り当てられている IP アドレスを示します。
Properties	RDU に設定されているプロパティを示します。
Version	現在使用中の RDU ソフトウェアのバージョンを示します。
Up Time	RDU が最後にダウンしてから、動作可能状態が継続している合計時間を示します。
State	RDU が要求に応答するかどうかを示します。管理者ユーザ インターフェイスに表示される唯一の状態は Ready です。
PACE Statistics	
Batches Processed	RDU が最後に起動してから処理された、個々のバッチの数を示します。
Batches Succeeded	RDU が最後に起動してから正常に処理された、個々のバッチの数を示します。
Batches Dropped	RDU が最後に起動してから破棄されたバッチの数を示します。
Batches Failed	RDU が最後に起動してから処理が失敗したバッチの数を示します。
Average Processing Time	RDU がビジーでキューに留まっていた時間を除いて、バッチの処理にかかった平均時間をミリ秒単位で示します。
Average Batch Processing Time	RDU がビジーでキューに留まっていた時間を含めて、バッチの処理にかかった平均時間をミリ秒単位で示します。
Configuration Regeneration Statistics	
State	構成生成サービスの動作状態を示します。次の状態があります。 • Idle：CRS が要求の再生成を処理しないことを示します。 • Regeneration：CRS が要求の再生成を処理することを示します。 • Waiting Regeneration：CRS がデバイスの構成を再生成できないことを示します。CRS がこの状態から先に進まない場合は、*rdu.log* で詳細を確認してください。
Requests Processed	RDU が最後に起動してから処理された構成再生成要求の数を示します。
Log Files	
RDU Log File	View Details アイコンが表示されます。このアイコンをクリックすると、View Log File Contents ページが表示され、*rdu.log* ファイルの詳細を確認できます。
Audit Log File	View Details アイコンが表示されます。このアイコンをクリックすると、View Log File Contents ページが表示され、*audit.log* ファイルの詳細を確認できます。

表 12-6　View Regional Distribution Unit Details ページ（続き）

フィールドまたはボタン	説明
Device Statistics	
（注） Device Statistics セクションは、該当するデバイスが存在する場合にのみ表示されます。	
	RDU データベース内のデバイスの数を示します。この領域に表示される情報は、ライセンスされ設定されたテクノロジーによって異なります。次のようなデバイスが含まれています。 • DOCSIS モデム • コンピュータ • PacketCable MTA • CableHome WAN-Data/WAN-MAN デバイス • STB
（注） JAR ファイルをインストールしている場合は、インストール済みの拡張 JAR ファイルとロードされた拡張クラス ファイルの情報が Device Statistics セクションの後に表示されます。	

CHAPTER 13

Broadband Access Center の設定

この章では、Configuration メニューでオプションを選択して行う、Broadband Access Center（BAC）の設定作業について説明します。この作業は次のとおりです。

サービス クラスの設定

BAC 管理者のユーザ インターフェイスを使用すると、お客様に提供するサービス クラスを設定できます。たとえば、DOCSIS オプションをさまざまな DOCSIS サービス クラスに関連付けることができます。BAC 管理者のユーザ インターフェイスを使用して、選択したサービス クラスを追加、修正、表示、または削除できます。

表 13-1 に、**Configuration > Class of Service > Manage Class of Service** をクリックしたときに表示されるフィールドおよびボタンを示します。

表 13-1　Manage Class of Service ページ

フィールドまたはボタン	説明
Class of Service	
Class of Service	検索できるテクノロジー サービス クラスを示すドロップダウン リストです。次のオプションを選択できます。 • CableHome WAN-Data • CableHome WAN-MAN • Computer • DOCSIS Modem • PacketCable Multimedia Terminal Adapter (MTA) • STB **(注)** これらのテクノロジー分野の詳細については、P.13-7 の「デフォルトの設定」を参照してください。
Class of Service	サービス クラス オブジェクトの名前が表示されます。

サービス クラスの追加

特定のサービス クラスを追加するには、次の手順に従います。

ステップ 1　Manage Class of Service ページの Class of Service ドロップダウン リストを使用して、サービス クラスを追加するデバイス タイプを選択します。

ステップ 2　**Add** をクリックします。

Add Class of Service ページが表示されます。このページでは、選択したサービス クラスの各種の設定を指定します。

ステップ 3　新しいサービス クラスの名前を入力し、Class of Service Type ドロップダウン リストからデバイス タイプを選択します。たとえば、Gold-Classic という名前の DOCSIS モデム用の新しいサービス クラスを作成するとします。その場合、Class of Service Name に **Gold-Classic** と入力し、サービス タイプ ドロップダウン リストから **DOCSISModem** を選択します。

ステップ 4　プロパティを選択し、それに対応する値を Property Value フィールドに入力します。たとえば、プロパティ名として */cos/docsis/file* を選択し、Property Value フィールドに **Gold-Classic.cm** と入力して、残りの手順を続行します。

（注） DOCSISModem サービス クラスを追加する場合は、前に追加したファイルの名前を値にして、*/cos/docsis/file* プロパティを指定する必要があります。このフィールドは、このサービス クラスを含む DOCSIS デバイスをプロビジョニングする場合に使用します。

BAC には、ケーブル モデム設定ファイルを自動的に選択する機能があり、これによって最上位 DOCSIS バージョンがモデムと互換性を持つことができるようになります。この機能を有効にするには、DOCSIS レベルごとに 1 つずつ、複数の設定ファイルを使用してサービス クラスを設定する必要があります。次のプロパティを使用して、DOCSIS バージョンに特定の設定ファイルを選択できるようにしてください。

- */cos/docsis/file/1.0*：DOCSIS 1.0 に特定の設定ファイルを選択します。
- */cos/docsis/file/1.1*：DOCSIS 1.1 に特定の設定ファイルを選択します。
- */cos/docsis/file/2.0*：DOCSIS 2.0 に特定の設定ファイルを選択します。
- */cos/docsis/file/3.0/ipv4*：IPv4 モードの DOCSIS 3.0 に特定の設定ファイルを選択します。
- */cos/docsis/file/3.0/ipv6*：IPv6 モードの DOCSIS 3.0 に特定の設定ファイルを選択します。

PacketCable サービス クラスを追加する場合は、前に追加したファイルの名前を値にして、*/cos/packetCableMTA/file* プロパティを指定する必要があります。このフィールドは、このサービス クラスを含む PacketCable デバイスをプロビジョニングする場合に使用します。

CableHome WAN-MAN サービス クラスを追加する場合は、前に追加したファイルの名前を値にして、*/cos/cableHomeWanMan/file* プロパティを指定する必要があります。このフィールドは、このサービス クラスを含む CableHome WAN-MAN デバイスをプロビジョニングする場合に使用します。

ステップ 5　**Add** をクリックして、サービス クラスにそのプロパティを追加します。

ステップ 6　**Submit** をクリックして手順を完了します。

サービス クラスを確定すると、Manage Class of Service ページが表示され、その特定のデバイス タイプで新規に追加されたサービス クラスが示されます。

サービス クラスの修正

サービス クラスを修正するには、種々のプロパティを選択し、適切なプロパティ値を割り当てます。サービス クラスを初めて作成する場合は、必要なプロパティをすべて選択し、値を割り当てる必要があります。入力内容に誤りがあった場合や、特定のサービス クラスを修正することが必要になった場合は、以前の修正を確定する前にプロパティ値を修正するか、または Property Name と Property Value のペアをまとめて削除します。

（注） サービス クラス オブジェクトに変更を加えると、影響を受けるすべてのデバイスの構成が Configuration Regeneration Service（CRS; 構成再生成サービス）によって再生成され、DPE に送信されます。CRS は、このタスクをバックグラウンド ジョブとして実行します。

CRS のステータスは、View RDU Details ページから表示できます。

サービス クラスのプロパティを追加、削除、または修正するには、次の手順に従います。

ステップ 1　Manage Class of Service ページから、特定のデバイス タイプのサービス クラスを選択します。

Modify Class of Service ページが表示されます。

- 選択したサービス クラスに新しいプロパティを追加するには、次の手順に従います。
  - Property Name ドロップダウン リストから、選択したサービス クラスに割り当てる最初のプロパティを選択し、そのプロパティの適切な値を選択して、**Add** をクリックします。
  - 選択したサービス クラスに割り当てる他のすべてのプロパティについて、この手順を繰り返します。
- 選択したサービス クラスのプロパティを削除するには、次の手順に従います。
  - Property Name ドロップダウン リストのすぐ上にあるリストで、不要なプロパティを見つけます。
  - **Delete** をクリックします。
- プロパティに現在割り当てられている値を修正するには、次の手順に従います。
  - 上記と同じ方法で、該当するプロパティを削除します。
  - 新しいプロパティ値を使用して同じプロパティをもう一度追加します。

（注）　業務に必須のプロパティを削除する場合は、変更を確定する前に、そのプロパティを再度追加して適切な値を選択してください。

ステップ 2　**Submit** をクリックします。

Submit をクリックすると、サービス クラスに追加された各プロパティが表示されます。次に、選択したサービス クラスでデバイスに対する構成を再生成するための確認ページが表示されます。

ステップ 3　**OK** をクリックします。

Manage Class of Service ページで、修正したサービス クラスが使用可能になります。

サービス クラスの削除

既存のサービス クラスはどれでも削除できます。ただし、削除する前に、そのサービス クラスに関連付けられたデバイスが存在しないことを確認してください。

ヒント　削除するサービス クラスに関連付けられたデバイスが多数存在する場合は、BAC Application Programming Interface（API; アプリケーション プログラミング インターフェイス）を使用して、これらすべてのデバイスに別のサービス クラスを再割り当てするプログラムを記述します。

（注）　デフォルトのサービス クラスとして指定されている場合、またはデバイスが関連付けられている場合は、そのサービス クラスは削除できません。したがって、**unprovisioned-docsis** サービス クラス オブジェクトは削除できません。デバイスが関連付けられているサービス クラスを削除しようとすると、次のエラー メッセージが表示されます。

```
The following error(s) occurred while processing your request.
Error: Class Of Service [sample-CoS] has devices associated with it, unable to delete

Please correct the error(s) and resubmit your request.
```

エラー メッセージでは、特定のサービス クラスが指定されます。この例では、*sample-CoS* を使用します。

サービス クラスを削除するには、次の手順に従います。

ステップ 1　Manage Class of Service ページから、削除する特定のデバイス タイプのサービス クラスを選択します。

ステップ 2　そのサービス クラスの **Delete** アイコン（🗑）をクリックします。

確認ダイアログボックスが表示されます。

ステップ 3　**OK** をクリックします。

カスタム プロパティの設定

カスタム プロパティを使用すると、RDU データベースに保存される追加のカスタマイズ可能なデバイス情報を指定できます。カスタム プロパティを設定するには、**Configuration > Custom Property > Manage BAC Custom Properties** をクリックします。このページを使用して、カスタム プロパティを追加または削除します。

注意　カスタム プロパティは使用中でも削除できますが、削除すると、そのプロパティを使用している他の領域に深刻な障害が起こる原因になります。

カスタム プロパティを定義すると、プロパティ階層で使用できるようになります。P.4-7 の「プロパティ階層」を参照してください。

また、プロパティは Node オブジェクトおよび Node Type オブジェクトでも設定できますが、それらのプロパティはプロパティ階層に含まれません。

カスタム プロパティの追加

カスタム プロパティを追加するには、次の手順に従います。

ステップ 1　Manage BAC Custom Properties ページから、**Add** をクリックします。

Add Custom Property ページが表示されます。

ステップ 2　新しいカスタム プロパティの名前を入力します。

ステップ 3　ドロップダウン リストに表示される選択肢から、カスタム プロパティのタイプを選択します。

ステップ 4　**Submit** をクリックします。

プロパティがデータベースに追加されると、Manage BAC Custom Properties ページが表示されます。

カスタム プロパティの削除

カスタム プロパティを削除するには、次の手順に従います。

ステップ 1　Manage BAC Custom Properties ページから削除するカスタム プロパティを指定します。

ステップ 2　カスタム プロパティに対応する **Delete** アイコンをクリックします。

確認ダイアログボックスが表示されます。

ステップ 3　**OK** をクリックします。

データベースからカスタム プロパティが削除された状態で Manage BAC Custom Properties ページが表示されます。

デフォルトの設定

Regional Distribution Unit（RDU）、Network Registrar 拡張、およびすべてのサポート テクノロジーなど、システム全体のデフォルト設定にアクセスできます。デフォルト設定を設定または表示するには、**Configuration > Defaults** をクリックします。**Configure Defaults** ページが表示されます。

特定のデフォルト ページにアクセスするには、画面左側の Default リンクから特定のリンクをクリックします。

この項では、次のトピックについて取り上げます。

CableHome WAN のデフォルト

CableHome WAN には 2 つの明確なデフォルト画面があり、1 つを WAN-Data デバイス用に、もう 1 つを WAN-MAN デバイス用に使用します。どちらの場合でも、左側のペインから適切なデフォルトを選択します。

- CH WAN-Data Defaults リンクを選択すると、CableHome WAN-Data Defaults ページが表示されます。このページを使用して WAN-Data デバイスを構成します。
- CH WAN-MAN Defaults リンクを選択すると、CableHome WAN-MAN Defaults ページが表示されます。このページを使用して WAN-MAN デバイス タイプを設定します。

各 WAN デフォルト ページには、表 13-2 に示されるフィールドと同一のフィールドがあります。

表 13-2　Configure Defaults-CH WAN-Data/CH WAN-MAN Defaults ページ

フィールドまたはボタン	説明
CableHome WAN-Data Defaults/CableHome WAN-MAN Defaults	
Extension Point	WAN デバイスの構成を生成するときに実行する拡張ポイントを示します。
Disruption Extension Point	WAN デバイスを中断するために実行される拡張ポイントを示します。
Service-level Selection Extension Point	デバイスで必要とされる DHCP 基準とサービス クラスを判別するために使用される拡張を示します。
Default Class of Service	WAN-Data の現在のデフォルト サービス クラスを示します。このサービス クラスには、新しい未認識の WAN デバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。
Default DHCP Criteria	特定のデバイス テクノロジーに対する現在のデフォルト DHCP 基準を示します。このデフォルト DHCP 基準には、新しい未認識の WAN デバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。

表 13-2　Configure Defaults-CH WAN-Data/CH WAN-MAN Defaults ページ（続き）

フィールドまたはボタン	説明
Automatic FQDN Generation	デバイスのホストおよびドメイン名を自動的に生成します。次の 2 つのオプションの中から選択できます。 • Enabled：FQDN の自動生成をイネーブルにします。 • Disabled：自動 FQDN 生成をディセーブルにします。 （注）詳細については、P.13-32 の「自動 FQDN 生成」を参照してください。

コンピュータのデフォルト

Computer Defaults リンクを選択すると、BAC によってサポートされているコンピュータに現在適用されているデフォルト値のリストが表示されます。このページに表示されるフィールドの説明については、表 13-2 を参照してください。

（注）デフォルト サービス クラスまたはデフォルト DHCP 基準に変更を加えると、再生成が行われます。このページでそれ以外の変更を行っても、既存のデバイスに影響を与えません。

DOCSIS のデフォルト

DOCSIS Defaults リンクを選択すると、BAC によってサポートされているケーブル モデムに現在適用されているデフォルト値のリストが表示されます。このページに表示されるすべてのフィールドとボタンの説明については、表 13-3 を参照してください。

表 13-3　Configure Defaults-DOCSIS Defaults ページ

フィールドまたはボタン	説明
Extension Point	DOCSIS デバイスの構成を生成するときに実行する拡張ポイントを示します。
Disruption Extension Point	DOCSIS デバイスを中断するために実行される拡張ポイントを示します。
Service-level Selection Extension Point	デバイスで必要とされる DHCP 基準とサービス クラスを判別するために使用される拡張を示します。
Default Class of Service	デバイスの現在のデフォルト サービス クラスを示します。このサービス クラスには、新しい未認識のデバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。
Default DHCP Criteria	特定のデバイス テクノロジーに対する現在のデフォルト DHCP 基準を示します。このデフォルト DHCP 基準には、新しい未認識のデバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。
TFTP Modem Address Option	TFTP モデム アドレス オプションがイネーブルかどうかを示します。
TFTP Time Stamp Option	TFTP サーバがタイムスタンプを発行するかどうかを示します。

表 13-3　Configure Defaults-DOCSIS Defaults ページ（続き）

フィールドまたはボタン	説明（続き）
（注）このページで TFTP オプションの一方または両方をイネーブルにすると、DOCSIS ケーブル モデムに送信される前に、適切な TFTP 情報が TFTP ファイルに書き込まれます。	
Automatic FQDN Generation	デバイスのホストおよびドメイン名を自動的に生成します。次の 2 つのオプションの中から選択できます。 • Enabled：FQDN の自動生成をイネーブルにします。 • Disabled：自動 FQDN 生成をディセーブルにします。 （注）詳細については、P.13-32 の「自動 FQDN 生成」を参照してください。
CMTS Shared Secret	設定ファイルでの CMTS MIC の計算で BAC が使用する文字列を示します。CMTS はその文字列を使用して、認可のためにケーブル モデムが CMTS に送信する設定ファイルを認証します。
CMTS Default DOCSIS Version	すべての CMTS が使用するデフォルトの DOCSIS バージョンを指定します。DOCSIS バージョンをこのフィールドに入力しない場合、デフォルト バージョンは 1.0 になります。
Relay Agent IP Address to CMTS Version Mapping file	CMTS が使用するマッピング ファイルを示します。このファイルにより、CMTS が使用する DOCSIS バージョンを指定します。

（注）デフォルト サービス クラスまたはデフォルト DHCP 基準に変更を加えると、再生成が行われます。TFTP オプションへの変更は、次の TFTP 転送の開始時点で有効になります。

Network Registrar のデフォルト

BAC には Cisco Network Registrar（NR）拡張ポイントが組み込まれており、これにより BAC は、着信 DHCP パケットから情報を引き出してデバイスのテクノロジーを検出できます。また拡張ポイントにより BAC は、DPE に保存されている設定に対応するオプションを選択してデバイスの DHCP 要求に応答することもできます。

NR Defaults リンクを選択すると、Network Registrar 拡張に現在適用されているデフォルト値のリストが表示されます。表 13-4 に、このページに表示されるフィールドを示します。

表 13-4　Configure Defaults-Network Registrar Defaults ページ

フィールドまたはボタン	説明
NR Extension Point Settings (BAC 2.6, 2.7)	
Attributes Required in Request Dictionary	構成を生成する要求を RDU に送信するときに Network Registrar 要求辞書に含める必要がある属性のリストをカンマ区切り形式で示します。
Attributes from Request Dictionary as Bytes	デバイス構成を生成する要求を RDU に送信するときに Network Registrar 要求辞書からバイトとして引き出される属性のリストをカンマ区切り形式で示します。

表 13-4　Configure Defaults-Network Registrar Defaults ページ （続き）

フィールドまたはボタン	説明
Attributes from Request Directory as Strings	デバイス構成を生成する要求を RDU に送信するときに、Network Registrar 要求辞書から文字列として引き出される属性のリストをカンマ区切り形式で示します。
NR Extension Point Settings (BAC 4.0)	
Attributes Required in DHCPv4 Request Dictionary	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv4 要求辞書に含める必要がある属性のリストをカンマ区切り形式で示します。 このフィールドのデフォルト値は、リレー エージェントのリモート ID オプションです。このフィールドで **relay-agent-remote-id** 値を設定しない場合、Network Registrar 拡張は、デバイスが構成生成の要求をトリガーすることを拒否します。
Attributes from DHCPv4 Request Dictionary as Bytes	デバイス構成を生成する要求を RDU に送信するときに、Network Registrar DHCPv4 要求辞書からバイトとして引き出される属性のリストをカンマ区切り形式で示します。
Attributes from DHCPv4 Request Dictionary as Strings	デバイス構成を生成する要求を RDU に送信するときに、Network Registrar DHCPv4 要求辞書から文字列として引き出される属性のリストをカンマ区切り形式で示します。
Attributes Required in DHCPv6 Request Dictionary	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv6 要求辞書に含める必要がある属性のリストをカンマ区切り形式で示します。 このフィールドのデフォルト値は **none** です。
Options Required in DHCPv6 Request Dictionary	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv6 要求辞書に含める必要がある DHCP オプションのリストをカンマ区切り形式で指定します。
Attributes from DHCPv6 Request Dictionary as Bytes	デバイス構成を生成する要求を RDU に送信するときに、Network Registrar DHCPv6 要求辞書からバイトとして引き出される属性のリストをカンマ区切り形式で示します。
Options from DHCPv6 Request Dictionary as Bytes	デバイス構成を生成する要求を RDU に送信するときに、Network Registrar DHCPv6 要求辞書からバイトとして引き出される DHCP オプションのリストをカンマ区切り形式で指定します。
Attributes Required in DHCPv6 Relay Dictionary	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv6 リレー辞書に含める必要がある属性のリストをカンマ区切り形式で示します。 このフィールドのデフォルト値は **peer-address** です。
Options Required in DHCPv6 Relay Dictionary	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv6 リレー辞書に含める必要がある DHCP オプションのリストをカンマ区切り形式で示します。
Attributes from DHCPv6 Relay Dictionary as Bytes	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv6 リレー辞書からバイトとして引き出される属性のリストをカンマ区切り形式で示します。
Options from DHCPv6 Relay Dictionary as Bytes	Network Registrar 拡張が要求を RDU に送信してデバイス構成を生成するために、Network Registrar DHCPv6 リレー辞書からバイトとして引き出される DHCP オプションのリストをカンマ区切り形式で示します。

表 13-4　Configure Defaults-Network Registrar Defaults ページ （続き）

フィールドまたはボタン	説明
NR Extension Point Environment Settings	
Attributes from Environment Dictionary	要求を RDU に送信してデバイス構成を生成するときに Network Registrar 環境辞書から文字列として引き出される属性のリストをカンマ区切り形式で示します。

（注）　このページでの変更は、Network Registrar 拡張がリロードされるまで有効になりません。

PacketCable のデフォルト

PacketCable Defaults ページには、PacketCable 音声テクノロジーをサポートするために必要なデフォルト値が示されます。PacketCable Defaults リンクを選択すると、PacketCable デバイスに現在適用されているデフォルト値のリストが表示されます。表 13-5 に、このデフォルト ページに固有のフィールドを示します。

表 13-5　Configure Defaults-PacketCable Defaults ページ

フィールドまたはボタン	説明
Extension Point	このテクノロジーのデバイスの構成を生成するときに実行する拡張ポイントを示します。
Disruption Extension Point	このテクノロジーを中断するために実行される拡張ポイントを示します。
Service-level Selection Extension Point	デバイスで必要とされる DHCP 基準とサービス クラスを判別するために使用される拡張を示します。
Default Class of Service	デバイスの現在のデフォルト サービス クラスを示します。このサービス クラスには、新しい未認識のデバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。
Default DHCP Criteria	特定のデバイス テクノロジーに対する現在のデフォルト DHCP 基準を示します。このデフォルト DHCP 基準には、新しい未認識のデバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。
SNMP Set Timeout	SNMP 設定タイムアウトを秒単位で示します。
MTA Provisioning Notification	MTA イベントが実行された通知。選択されたオプションに基づいて MTA がそのプロビジョニングの完全な通知を送信すると、イベントが発生します。次のオプションを選択できます。 • On Failure • On Success • During Provisioning • Always • Never
Automatic FQDN Generation	Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）が生成されるかどうかを示します。

RDU のデフォルト

RDU Default リンクを選択すると、その RDU で設定したデフォルト設定が表示されます。このページを使用して RDU を設定し、Network Registrar と通信します。詳細については、『*User Guide for Cisco Network Registrar 7.0*』を参照してください。

表 13-6 に、RDU Defaults ページに表示されるフィールドの説明を示します。

表 13-6　Configure Defaults-RDU Defaults ページ

フィールドまたはボタン	説明
Configuration Extension Point	他のテクノロジー拡張ポイントが実行される前に実行する共通拡張ポイントを指定します。
Device Detection Extension Point	デバイスの DHCP 検出要求から引き出される情報に基づいてデバイス タイプ（たとえば、DOCSIS やコンピュータ）を判別するために使用される拡張ポイントを示します。
Publishing Extension Point	RDU パブリッシング プラグインに使用される拡張ポイントを示します。この情報は、RDU データを別のデータベースにパブリッシングするときに役立ちます。
Extension Point JAR File Search Order	上記の 4 つのフィールドにリストされている JAR ファイルでクラスを検索するときの順序を指定します。
CCM Server IP Address	CCM サーバの IP アドレスを示します。
CCM Server Port	BAC が通信で使用する CCM サーバ ポートを示します。
CCM Server User	CCM サーバのユーザ名を示します。ユーザ名は、パスワードフィールドと併用されます。
CCM Server Password	CCM サーバ ユーザの認証で使用されるパスワードを示します。
CCM Server Confirm Password	CCM サーバ パスワードを確証します。
CCM Server	CCM サーバへの BAC インターフェイスをイネーブルにするか、またはディセーブルにするかを指定します。
CCM Server Timeout	接続が切断されていると宣言するまでに、BAC が CCM サーバへの接続を試みる時間を秒単位で指定します。
CRS	Configuration Regeneration Service（CRS; 構成再生成サービス）がイネーブルかどうかを示します。次の 2 つのオプションがあります。 • Enable：BAC 内の CRS をイネーブルにします。 • Disable：BAC 内の CRS をディセーブルにします。

（注） RDU 拡張ポイントの詳細については、P.13-27 の「RDU 拡張の管理」を参照してください。

システム デフォルト

Systems Defaults リンクを選択すると、System Defaults ページが表示されます。表 13-7 に、このページに表示されるフィールドの説明を示します。

（注） デフォルト値は、BAC API を使用して設定できます。

表 13-7　System Defaults の Configure Defaults ページ

フィールドまたはボタン	説明
System Defaults	
SNMP Write Community String	SNMP 情報を必要とする可能性があるすべてのデバイスのデフォルト ライト（write）コミュニティ ストリングを示します。デフォルトのライト（write）コミュニティ ストリングは **private** です。
SNMP Read Community String	SNMP MIB を読み取りまたはアクセス可能なすべてのデバイスのデフォルト リード（read）コミュニティ ストリングを示します。デフォルトのリード（read）コミュニティ ストリングは **public** です。
Default Device Type for Device Detection	これまで RDU に登録されていないデバイスのデフォルトのデバイス タイプを指定します。次のオプションがあります。 • DOCSIS • COMPUTER • PacketCableMTA • STB • CableHomeWanMan • CableHomeWanData • None **（注）** デバイス検出拡張がデバイス　タイプを特定できない場合、デバイス タイプは「デフォルト タイプ」（たとえば、COMPUTER）に指定されます。デフォルトのデバイス タイプを None に設定した場合、デバイス レコードは RDU に追加されません。
Maximum Diagnostics Device Count	一度にトラブルシューティングを行うことができる MAC アドレス（デバイス）の最大数を示します。
MIB List	RDU を再起動する必要のない、RDU が使用する MIB のリストを示します。
Supplemental MIB List	RDU が使用する MIB の拡張リストを示します。
Excluded MIB Tokens	MIB によって再定義できないキーワードまたはトークンを定義します。
Excluded Supplemental MIB Tokens	MIB によって再定義できず、Excluded MIB Tokens リストに表示されない、追加キーワードまたはトークンを定義します。
Promiscuous Policy Settings	
CableHome WanData Promiscuous Mode	無差別モードの CableHome WAN-Data デバイスをイネーブルまたはディセーブルにします。
CableHome WanMan Promiscuous Mode	無差別モードの CableHome WAN-MAN デバイスをイネーブルまたはディセーブルにします。
Computer Promiscuous Mode	無差別モードのコンピュータをイネーブルまたはディセーブルにします。
PacketCable Promiscuous Mode	無差別モードの PacketCable デバイスをイネーブルまたはディセーブルにします。
STB Promiscuous Mode	無差別モードの STB をイネーブルまたはディセーブルにします。

表 13-7　System Defaults の Configure Defaults ページ （続き）

フィールドまたはボタン	説明
System Defaults	
CableHome WanData Promiscuous DHCP Criteria	無差別モードで WAN-Data デバイスをプロビジョニングするために使用する DHCP 基準を示します。
CableHome WanMan Promiscuous DHCP Criteria	無差別モードで WAN-MAN デバイスをプロビジョニングするために使用する DHCP 基準を示します。
Computer Promiscuous DHCP Criteria	無差別モードでコンピュータをプロビジョニングするために使用する DHCP 基準を示します。
Packetcable Promiscuous DHCP Criteria	無差別モードで PacketCable デバイスをプロビジョニングするために使用する DHCP 基準を示します。
STB Promiscuous DHCP Criteria	無差別モードで STB をプロビジョニングするために使用する DHCP 基準を示します。
CableHome WanData Promiscuous Class of Service	無差別モードで WAN-Data デバイスをプロビジョニングするために使用するサービス クラスを示します。
CableHome WanMan Promiscuous Class of Service	無差別モードで WAN-MAN デバイスをプロビジョニングするために使用するサービス クラスを示します。
Computer Promiscuous Class of Service	無差別モードでコンピュータをプロビジョニングするために使用するサービス クラスを示します。
Packetcable Promiscuous Class of Service	無差別モードで PacketCable デバイスをプロビジョニングするために使用するサービス クラスを示します。
STB Promiscuous Class of Service	無差別モードで STB をプロビジョニングするために使用するサービス クラスを示します。

STB のデフォルト

STB Defaults ページには、特に RNG-200 STB など、進歩する OpenCable Application Platform に基づいたビデオ セットトップ ボックス（STB）をサポートするために必要なデフォルト値が示されます。表 13-8 に、このページに表示されるフィールドの説明を示します。

表 13-8　Configure Defaults-STB Defaults ページ

フィールドまたはボタン	説明
Extension Point	STB の設定を生成するときに実行する拡張ポイントを示します。
Disruption Extension Point	STB を中断するために実行される拡張ポイントを示します。
Service-level Selection Extension Point	デバイスで必要とされる DHCP 基準とサービス クラスを判別するために使用される拡張を示します。
Default Class of Service	STB の現在のデフォルト サービス クラスを示します。このサービス クラスには、新しい未認識の STB デバイスが割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。
Default DHCP Criteria	特定のデバイス テクノロジーに対する現在のデフォルト DHCP 基準を示します。このデフォルト DHCP 基準には、新しい未認識の STB が割り当てられます。ドロップダウン リストを使用して新しいデフォルト値を選択してください。

表 13-8　Configure Defaults-STB Defaults ページ（続き）

フィールドまたはボタン	説明
Automatic FQDN Generation	デバイスのホストおよびドメイン名を自動的に生成します。次の 2 つのオプションの中から選択できます。 • Enabled：FQDN の自動生成をイネーブルにします。 • Disabled：自動 FQDN 生成をディセーブルにします。 （注）詳細については、P.13-32 の「自動 FQDN 生成」を参照してください。

（注）以降のデバイス構成には、ここで実装する変更が含まれます。ただし、既存の構成はすべて変更されません。既存の構成で変更を行う場合は、API を使用して構成を再生成する必要があります。

DHCP 基準の設定

BAC では、DHCP 基準として、Network Registrar でスコープを選択する場合のデバイスに特有の基準を記述します。たとえば、**provisioned-docsis** という DHCP 基準には、**tagProvisioned** という選択タグが含まれています。DHCP 基準は DOCSIS モデムと関連付けられます。このモデムが Network Registrar から IP アドレスを要求すると、Network Registrar はスコープ選択タグ **tagProvisioned** に関連付けられたスコープを探します。

DHCP Criteria ページにアクセスするには、**Configuration > DHCP Criteria** を選択します。追加したテクノロジー DHCP 基準を識別する DHCP 基準のリストが、Manage DHCP Criteria ページに表示されます。

DHCP 基準の追加

DHCP 基準を追加するには、次の手順に従います。

ステップ 1　Manage DHCP Criteria ページで **Add** をクリックします。

Add DHCP Criteria ページが表示されます。

ステップ 2　作成する DHCP 基準の名前を入力します。

ステップ 3　DHCP 基準クライアント クラス名を入力します。

ステップ 4　包含選択タグと除外選択タグを入力します。

（注）新しい DHCP 基準を作成するときには、入力するクライアント クラス名およびインクルード選択タグと除外選択タグ名を、Network Registrar 内からの名前とまったく同じにする必要があります。クライアント クラスと選択タグの詳細については、*/docs* ディレクトリの『*User Guide for Cisco Network Registrar 7.0*』および *CLIFrame.html* を参照してください。新しい DHCP 基準の作成時には、クライアント クラスの名前またはインクルード選択タグと除外選択タグ名のいずれかを指定してください。

ステップ 5　DHCP 基準に追加されるプロパティを追加できます。Property Name を選択し、適切な Property Value を入力します。

ステップ 6　**Add** をクリックします。

ステップ 7　**Submit** をクリックします。

DHCP 基準が RDU データベースに正常に追加されると、Manage DHCP Criteria ページに表示されます。

DHCP 基準の修正

（注）　DHCP 基準を変更すると、実装する変更点がその後のデバイス構成に組み込まれます。ネットワークのデバイスはリブートされるまで新しい構成を取得しませんが、すべての既存の構成が再生成されます。

既存の DHCP 基準を修正するには、次の手順に従います。

ステップ 1　Manage DHCP Criteria ページで、修正する DHCP Criteria リンクをクリックします。

Modify DHCP Criteria ページが表示されます。

ステップ 2　クライアント クラス、インクルード選択タグと除外選択タグ、およびプロパティ値の設定に変更を加えます。

ステップ 3　**Submit** をクリックします。

RDU データベースで DHCP 基準の修正が正常に終了すると、Manage DHCP Criteria ページが表示されます。

DHCP 基準の削除

管理者アプリケーションを使用して DHCP 基準を削除しても、DHCP サーバから実際の DHCP サーバ設定は削除されません。DHCP サーバ設定を手動で削除する必要があります。

既存の基準を削除するには、次の手順に従います。

（注）　DHCP 基準は、デバイスが何も関連付けられておらず、デフォルトの DHCP 基準として指定されていない場合にのみ、削除することができます。DHCP 基準にデバイスが割り当てられている場合は、その DHCP 基準を削除する前に別の DHCP 基準と関連付ける必要があります。

ステップ 1　Manage DHCP Criteria ページで、削除する DHCP 基準に対応する **Delete** アイコンをクリックします。

確認ダイアログボックスが表示されます。

ステップ 2　**OK** をクリックします。

Manage DHCP Criteria ページが表示されます。

ファイルの管理

BAC 管理者のユーザ インターフェイスを使用して、DOCSIS、PacketCable MTA、および WAN-MAN ファイルの動的生成で使用する TFTP サーバ ファイルやテンプレート ファイル、またはデバイスのソフトウェア イメージを管理できます。このページを使用して、次に示すいずれかのファイル タイプを追加、削除、置換、またはエクスポートできます。

- テンプレート ファイル：DOCSIS、PacketCable、または CableHome オプションと値を含むテキスト ファイルです。特定のサービス クラスと一緒に使用することにより、ファイルを動的に生成できます。

（注） テンプレート ファイルは任意のテキスト エディタで作成できますが、ファイル タイプが *.tmpl* になっている必要があります。テンプレートの詳細については、P.5-2 の「テンプレート ファイル：概要」を参照してください。

- 静的設定ファイル：これらのファイルは、デバイスの設定ファイルとして使用されます。たとえば、*gold.cm* という静的設定ファイルは、Gold DOCSIS サービス クラスを示します。BAC は、このファイル タイプをその他のバイナリ ファイルと同様に扱います。
- ファームウェア イメージ：デバイス ファームウェアのイメージです。機能をアップグレードするために、デバイスにダウンロードできます。BAC は、このファイル タイプをその他のバイナリ ファイルと同様に扱います。これらのファームウェア イメージには、シスコ デバイスの IOS イメージを含めることもできます。

表 13-9 に、View Files ページに表示されるフィールドの説明を示します。

表 13-9　View Files ページ

フィールドまたはボタン	説明
Search Type	BAC 管理者のユーザ インターフェイスを使用し、ファイルに対して実行可能な検索のタイプを示します。次のオプションがあります。 • Search by File Name：指定するファイル名パターンを使用してファイルを検索します。 • Search by File Type：指定するファイル タイプを使用してファイルを検索します。次のオプションがあります。 — Firmware File：ファームウェア イメージ ファイルを指定します。 — CableLabs Configuration File：CableLabs の静的設定ファイルを指定します。 — CableLabs Configuration Template：CableLabs の設定テンプレート ファイルを指定します。 — Generic File：汎用ファイルを指定します。 — JAR File：JAR ファイルを指定します。 — MIB File：MIB ファイルを指定します。
Search Criteria	ファイル名またはファイル タイプを示します。アスタリスク（*）をワイルドカード文字として使用し、部分的なファイル名を検索できます。たとえば、***.cm** と入力して、*.cm* 拡張子で終わるすべてのファイルを一覧表示できます。bronze* は無効なワイルドカードの一例です。
Page Size	1 ページ内に表示させる必要がある結果数を示します。

表 13-9　View Files ページ （続き）

フィールドまたはボタン	説明
Files	検索基準と一致したファイルのリストが表示されます。 **（注）** このリストで選択した項目を削除するには、その項目のすぐ左にあるチェックボックスをオンにする必要があります。
View	選択したバイナリ ファイルの詳細情報が表示されます。
File Type	ファイルのタイプを示します。
Export	選択したファイルをクライアントのコンピュータにエクスポートします。

ファイルの追加

既存のファイルを追加するには、次の手順に従います。

ステップ 1　View Files ページで **Add** をクリックします。

Add Files ページが表示されます。

ステップ 2　ドロップダウン リストから File Type を選択します。

ステップ 3　ソース ファイルへのパスを入力します。

ソース ファイルの正確な名前がわからない場合は、**Browse** を使用して目的のディレクトリまで移動し、そのファイルを選択します。

ステップ 4　ファイルの名前を入力します。

CableLabs Configuration Template または Firmware File を追加する場合は、次の手順も実行する必要があります。追加しない場合はステップ 6 に進んでください。

a. CableLabs Configuration Template または Firmware File を追加する場合は、RDU に追加するファイルを DPE に配信できます。配信するには、Is Deliverable フィールドに対応する **Enabled** オプション ボタンをクリックします。

BAC がファイル タイプごとに配信可能ステータスを設定する間は、CableLabs Config Template または Firmware File の場合のみデフォルト設定を変更できます。次のリストに、ファイル タイプごとのデフォルトの配信可能ステータスの説明を示します。

- Firmware File：Enabled
- CableLabs Configuration File：Enabled
- CableLabs Configuration Template：Disabled
- Generic File：Disabled
- JAR File：Disabled
- MIB File：Disabled

b. Firmware File の場合は、ファイル バージョンとそのバージョンの適切な説明も追加入力できます。

ステップ 5　**Submit** をクリックします。

（注）　最大 4 MB のファイル サイズがサポートされています。追加するファイルのサイズが 4 MB を超えるとエラーが表示されます。

View Files ページが表示され、追加されたファイルが示されます。

ファイルの表示

DOCSIS または PacketCable 音声テクノロジー ファイルの内容を表示するには、次の手順に従います。

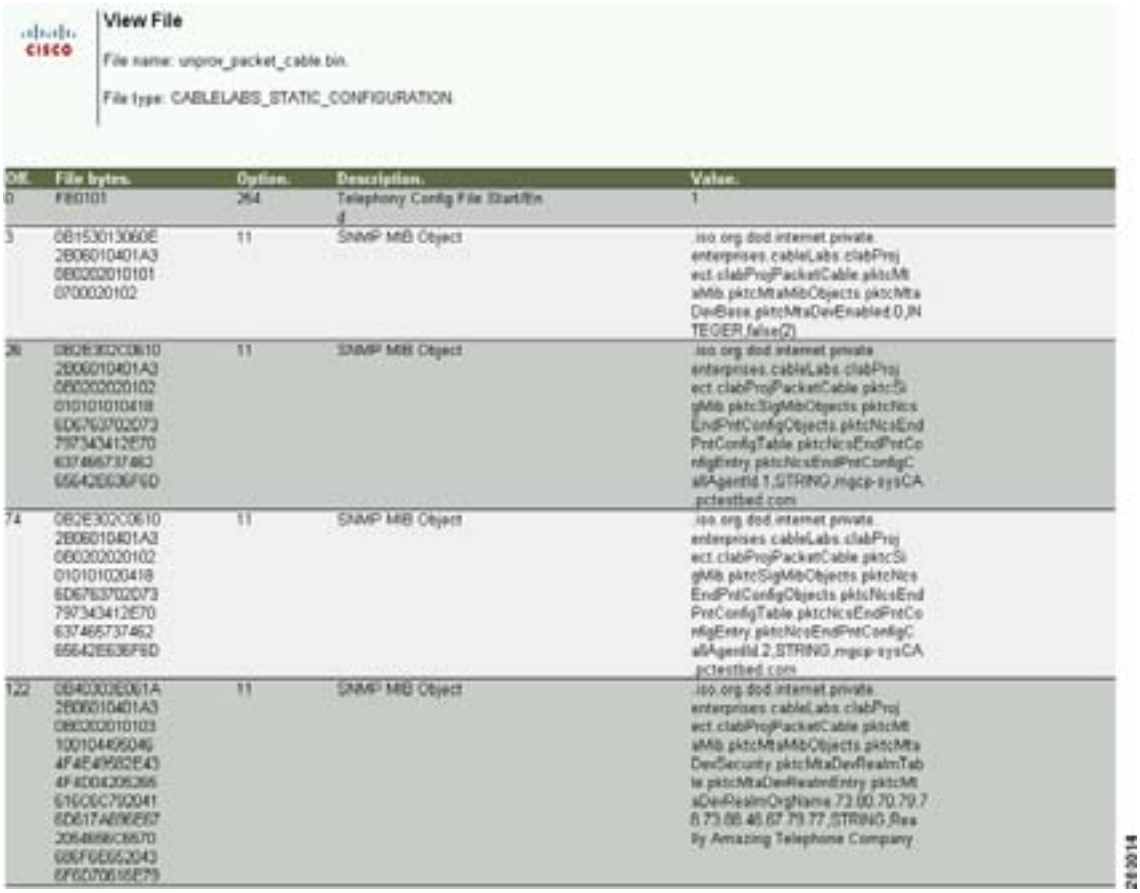
ステップ 1　View Files ページで、ファイル タイプまたはファイル名の検索オプションを使用して必要なファイルを検索します。

ステップ 2　そのファイルに対応する **View Details**（）アイコンをクリックします。

ファイルの内容の詳細を示した View File ページが表示されます。

図 13-1 に、サンプル バイナリ ファイルの内容を示します。

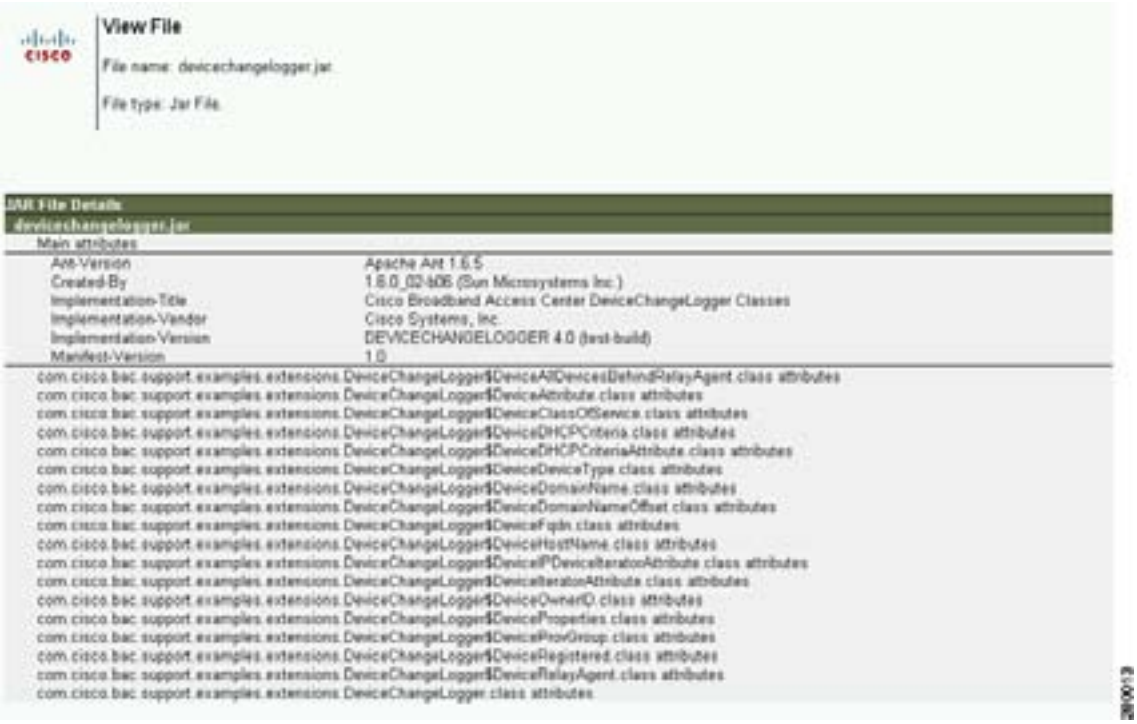
図 13-1　サンプル バイナリ ファイルの内容



（注）　この図に示される出力は省略されています。

図 13-2 に、サンプル JAR ファイルの内容を示します。

図 13-2　サンプル JAR ファイルの内容



ファイルの置換

既存のファイルを置換するには、次の手順に従います。

ステップ 1　View Files ページで、置換するファイルに対応する Files リンクをクリックします。

Replace File ページが表示されます。選択したファイル名がすでにこのページに表示されています。

ステップ 2　RDU データベースに存在するファイルと置換する、ソース ファイルのパスとファイル名を入力します。ソース ファイルの正確な名前や場所がわからない場合は、**Browse** を使用して目的のディレクトリまで移動し、そのファイルを選択します。

ステップ 3　**Submit** をクリックします。

置換ファイルを決定した後に確認ページが表示されて、置換後、影響を受けるデバイス用の構成が BAC によって再生成されることが示されます。

ステップ 4　**OK** をクリックします。

View Files ページが表示されます。

サービス クラスによってこのファイルを使用するすべてのデバイスの構成は、置換の完了後に再生成されます。

ファイルのエクスポート

エクスポート機能を使用して、ファイルを自分のローカル ハード ドライブにコピーできます。

（注）　次に示す手順は、Internet Explorer を使用している場合のものです。Netscape Navigator を使用している場合は、手順が異なります。

ファイルをエクスポートするには、次の手順に従います。

ステップ 1　View Files ページで、エクスポートするファイルに対応する Files リンクをクリックします。

ステップ 2　エクスポートするファイルを指定します。

ステップ 3　**Export** アイコン（ ）をクリックします。

ファイルを開くか、または保存するよう求めるメッセージが表示されます。

ステップ 4　BAC ユーザ インターフェイスに戻ります。

ファイルの削除

既存のファイルを削除するには、次の手順に従います。

ステップ 1　View Files ページで検索オプションを使用して、削除するファイルを指定します。

該当のファイルが Files リストに表示されます。

ステップ 2　該当の 1 つまたは複数のファイルを選択します。

ステップ 3　**Delete** をクリックします。

注意　サービス　クラスに直接リンクされていないが、サービス　クラスにリンクされている他のテンプレート ファイルによって参照されるテンプレート ファイルを削除すると、構成再生成サービスが失敗する原因になります。

（注）　サービス　クラスが関連付けられているファイルは削除できません。操作を続ける前に、サービス クラスの関連付けを解除する必要があります。詳細については、P.13-2 の「サービス クラスの設定」を参照してください。

ライセンスの管理

ソフトウェア ライセンスは、特定の機能を有効にするか、または自分の環境の機能を高めるために使用します。この項では、異なるライセンスの BAC での処理方法を説明します。このリリースでのライセンスの変更、およびライセンス ファイルの取得方法の詳細については、『*Release Notes for Cisco Broadband Access Center 4.0*』を参照してください。

BAC ライセンスは、永久ライセンスまたは評価ラインセンスのいずれかとして入手できます。

- Permanent：永久ライセンスは、自分のネットワーク環境で使用するために購入するライセンスで、それに対応する特定の機能が有効になります。
- Evaluation：評価ライセンスでは、機能が一定期間有効になります。

注意　評価ライセンスを使用して完全運用のネットワークへの展開を行わないようにしてください。評価ライセンスの期限が切れると、BAC を使用してネットワークのデバイスをプロビジョニングすることができなくなります。

BAC 評価ライセンスは、あらかじめ定められた日に無効になります。そのため評価ライセンスは、必要が生じたときに作成する必要があります。評価ライセンスを作成するには、シスコの担当者にお問い合せください。担当者は、必要なライセンス ファイルをオンラインで生成し、電子メール経由で転送します。

この BAC リリースでは、サービス ファイルを使用してライセンスできます。これらのライセンスにより、BAC を使用する一定数のサービスをプロビジョニングできます。各サービスは、システムでプロビジョニングされる 3 つの IP アドレスに変換されるため、10,000 サービス ライセンスは 30,000 個の IP アドレスに相当します。受信するライセンス ファイルには、購入したサービスの数ではなく、ライセンスが与えられている IP アドレスの数が書き込まれています。

注意　ライセンス　ファイルは編集しないでください。どのような方法であっても、データを変更するとライセンス ファイルが無効になります。

サービス ライセンスを使用すると、ライセンス ファイルに記載されている最大数までの範囲で、任意のデバイス タイプを任意の組み合せでプロビジョニングできます。このリリースの BAC でサポートされるデバイス タイプは次のとおりです。

- DOCSIS ケーブル モデム
- PacketCable MTA
- CableHome WAN-MAN デバイスと WAN-Data デバイス
- コンピュータ

- カスタム CPE

（注） 次の BAC コンポーネントについては、別個のライセンスが必要になります。

- DPE
- KDC（音声テクノロジーをサポートするようにネットワークを設定する場合）

DPE ライセンスは管理者のユーザ インターフェイスからインストールする必要がありますが、KDC ライセンスは前の BAC リリースの場合と同様に専用ライセンスのままであり、BAC のインストール中にライセンスが与えられます。

評価ライセンスを複数インストールすることはできませんが、期限切れやまだ有効な評価ライセンスであっても（それらより有効期限が後の）新しい評価ライセンスまたは永久ライセンスと置き換えることができます。既存のライセンスを置き換える場合には、新規ライセンスのデバイス制限が、最低でもデータベースに現在保存されているデバイスの数と同じになるようにしてください。

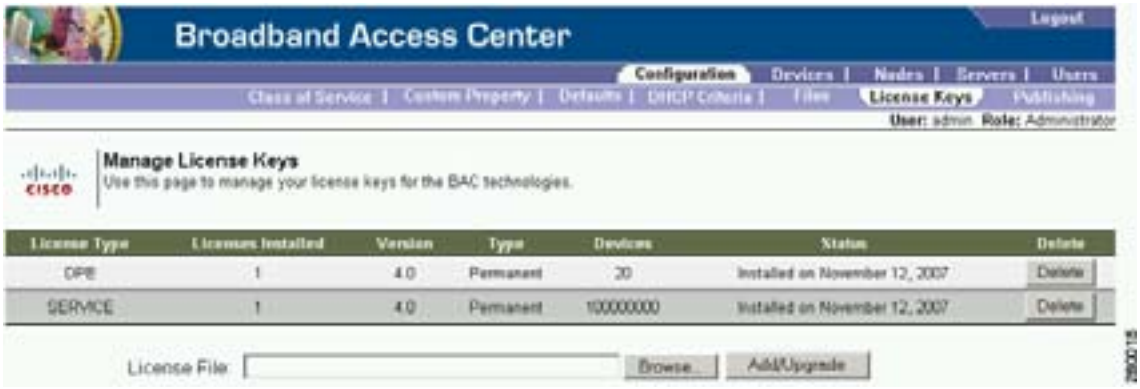
評価ライセンスから永久ライセンスにアップグレードするときに、ソフトウェアを再インストールしたり、BAC を再設定したりする必要はありません。BAC 管理者のユーザ インターフェイスを使用して、永久ライセンスをインストールするだけです。

（注） 永久ライセンスをすでにインストールしている場合は、評価ライセンスをインストールできません。

ライセンスを追加することで、永久ライセンスをアップグレードして、ライセンスされるデバイスの数を増やすことができます。ライセンスされたデバイスの数が上限に達すると、新しいデバイスをプロビジョニングできませんが、すでにプロビジョニングされた既存のデバイスは引き続きサービスを受けられます。

図 13-3 にサンプル Manage License Keys ページを示します。このページには、実装用に入力されたサービス ライセンスのリストが表示されます。

図 13-3　Manage License Keys ページ



ライセンスの追加と修正

『*Release Notes for Cisco Broadband Access Center 4.0*』で説明される要求プロセスに従って新しいライセンス ファイルを入手します。ライセンス ファイルを受信した後は、BAC 管理者のユーザ インターフェイスを起動する予定のシステムに各ファイルを保存します。

注意　ライセンス ファイルは編集しないでください。どのような方法であっても、データを変更するとライセンス ファイルが無効になります。

ライセンスを追加、修正、またはアップグレードするには、次の手順に従います。

ステップ 1　**Configuration** > **License Keys** の順に選択します。

（注）　ライセンスを初めてアップロードする場合は、Main Menu に表示されるライセンス リンクを使用できます。

Manage License Keys ページが表示されます。

ステップ 2　License File フィールドに、ローカル システムでのライセンス ファイルの場所への完全パスを入力します。

または、**Browse** をクリックしてライセンス ファイルの場所まで移動します。パス名を指定するときには、ライセンス ファイルの名前も忘れずに含めてください。

ステップ 3　**Add/Upgrade** をクリックします。

ライセンスが与えられているサービスおよび DPE の数に関する詳細が表示されます。

ライセンスの削除

評価または永久のどちらのであっても、Manage License Keys ページに表示されるライセンスを選択して削除できます。

RDU は、次の場合に評価ライセンスを自動的に削除します。

- 評価ライセンスがすでにインストールされている RDU に別の評価ライセンスをインストールする場合。追加する評価ライセンスが有効で、既存のライセンスが期限切れになった後に期限切れになる場合、インストール済みのライセンスは新しいライセンスで置き換えられます。
- 現在評価ライセンスを持っているところに永久ライセンスをアップロードする場合。永久ライセンスが有効な場合、評価ライセンスはその永久ライセンスによって置き換えられます。

（注）　削除することによってシステムでのライセンスの総許容量がシステムでプロビジョニングされているデバイスの数を下回る場合は、ライセンスを削除できません。

ライセンスを削除するには、次の手順に従います。

ステップ 1　**Configuration** > **License Keys** の順に選択します。

Manage License Keys ページが表示されます。

ステップ 2　削除するライセンスに対応する **Delete** ボタンをクリックします。

確認ダイアログボックスが表示されます。

ステップ 3　ライセンスの削除を確認するには、**Yes** をクリックします。

Manage License Keys ページにライセンス キーが表示されなくなります。

（注）　ライセンスが削除されたことを確認するには、そのアクションが *audit.log* に記録されているかどうかを調べてください。

RDU 拡張の管理

カスタム拡張ポイントの作成はプログラミング作業です。BAC 管理者のユーザ インターフェイスと併用することで、この作業では、BAC の動作を強化したり、新しいデバイス テクノロジーのサポートを追加したりできます。

拡張の管理方法を知る前に、BAC で必要とされる RDU 拡張ポイントについて理解してください。バッチのためにデバイスを中断する場合は、関連付けられたテクノロジーの中断拡張ポイントに少なくとも 1 つの中断拡張を添付する必要があります。

表 13-10 に、拡張を実行するために BAC で必要とされる RDU 拡張ポイントのリストを示します。

表 13-10　必要な RDU 拡張ポイント

拡張ポイント	説明	使用するかどうか	テクノロジーに特定かどうか
共通構成生成	デバイスの構成を生成するために実行されます。この拡張ポイントに添付される拡張は、テクノロジー固有のサービス レベル選択拡張の後、テクノロジー固有の構成生成拡張の前に実行されます。このリリースに組み込まれているデフォルト拡張では、この拡張ポイントを使用しません。	オプション	いいえ
構成生成	デバイスの構成を生成するために実行されます。	必須	はい
デバイス検出	デバイスの DHCP 検出要求パケットの情報に基づいて、デバイス テクノロジーを判断するために実行されます。	必須	いいえ
中断	デバイスを中断するために実行されます。	オプション	はい
パブリッシング	プロビジョニング データを外部データストアにパブリッシングするために実行されます。BAC に組み込まれるデフォルト拡張には、パブリッシング プラグインは含まれません。	オプション	いいえ
サービス レベル選択	デバイスに付与するサービス レベルを選択するために実行されます。この拡張ポイントに添付される拡張は、すべての共通構成生成拡張およびテクノロジー固有の設定生成拡張の前に実行されます。	オプション	はい

拡張の管理には、次の作業があります。

- 新しいクラスの作成（P.13-28）
- RDU カスタム拡張ポイントのインストール（P.13-29）
- RDU 拡張の表示（P.13-29）

（注） 拡張ポイントをカンマ区切りリスト形式で指定することにより、複数の拡張ポイントを指定できます。

新しいクラスの作成

次の手順は、カスタム拡張の作成プロセス全体をより明確に説明するためのものです。さまざまなタイプの拡張を作成できます。次の手順では、新しいパブリッシング拡張ポイントを使用します。

新しいクラスを作成するには、次の手順に従います。

ステップ 1　カスタム パブリッシング拡張に関する 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　拡張クラスを記述する 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 ファイルの作成方法およびコマンドライン JAR ツールの使用方法の詳細については、Java のドキュメント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 マニフェストには、名前と値のペアとしてフォーマットされた属性が含まれており、パッケージのバージョン情報を提供する属性のグループをサポートします。BAC はこの情報を含まない拡張 JAR ファイルを受け入れますが、ファイルのマニフェストにバージョン情報を含めてカスタム RDU 拡張をトラッキングすることをお勧めします。

マニフェスト情報は、**Servers > RDU > View Regional Distribution Unit Details** ページを選択して、管理者のユーザ インターフェイスから表示できます。インストール済みの拡張 JAR ファイルとロードされた拡張クラス ファイルの詳細情報は、Device Statistics セクションの後に表示されます。マニフェスト情報は RDU ログでも確認できます。

ステップ 3　カスタム拡張ポイントに関する 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 ファイルに任意の名前を付けることができます。わかりやすい名前を付けることができますが、既存の JAR ファイル名と重複しないようにしてください。

RDU カスタム拡張ポイントのインストール

JAR ファイルを作成したら、管理者のユーザ インターフェイスを使用してファイルをインストールします。

ステップ 1　P.13-19 の「ファイルの追加」を参照して、新しい JAR ファイルを追加します。

（注） JAR ファイル タイプを選択します。Browse 機能を使用して、P.13-28 の「新しいクラスの作成」で説明される手順に従って作成した JAR ファイルを指定し、このファイルをソースファイルとして選択します。File Name を空白にすると、ソースとファイルの両方に同じファイル名が割り当てられます。このファイル名が、管理者のユーザ インターフェイスに表示されます。

ステップ 2　**Submit** をクリックします。

ステップ 3　RDU Defaults ページに戻り、新しく追加された JAR ファイルが Extension Point JAR File Search Order フィールドに表示されるかどうかを確認します。

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

（注） そのクラス名が JAR ファイル内に存在しない場合、RDU はエラーを返します。このエラーは、主に JAR ファイルを置換するときに発生します。たとえば、設定したクラスが置換 JAR ファイル内で見つからない場合などです。

ステップ 5　**Submit** をクリックして、変更を RDU データベースにコミットします。

ステップ 6　RDU 拡張を表示し、正しい拡張がロードされることを確認します。

RDU 拡張の表示

すべての RDU 拡張の属性は、View Regional Distribution Unit Details ページに直接表示できます。このページには、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 ファイルに関する詳細が表示されます。P.12-31 の「Regional Distribution Unit の詳細の表示」を参照してください。

プロビジョニング データのパブリッシング

BAC には、追跡したプロビジョニング データを外部データストアにリアルタイムにパブリッシングする機能があります。そのためには、目的のデータストアにデータを書き込むパブリッシング プラグインを開発する必要があります。Manage Publishing ページには、プラグインの名前、その現在のステータス（イネーブルかどうか）、およびイネーブルまたはディセーブルにするためのスイッチが表示されます。

実装で必要とされる数のプラグインはすべてイネーブルにすることができますが、パブリッシング プラグインを使用するとシステム パフォーマンスが低下することを忘れないでください。

（注） BAC にはパブリッシング プラグインが付属していません。管理者は自分でプラグインを作成し、それらを JAR ファイルと同じ方法で BAC にロードする必要があります（P.13-19 の「ファイルの追加」を参照してください）。その後、Manage Publishing ページからプラグインを管理します。

データストアの変更のパブリッシング

パブリッシング プラグインをイネーブルまたはディセーブルにするには、次の手順に従います。

ステップ 1　プライマリ ナビゲーション バーの **Configuration** を選択します。

ステップ 2　セカンダリ ナビゲーション バーの **Publishing** を選択します。

Manage Publishing ページが表示されます。このページには、使用可能なデータベースのすべてのプラグインのリストが表示され、それぞれのプラグインの現在のステータスが示されます。

ステップ 3　目的のプラグインをイネーブルまたはディセーブルにするには、対応するステータス インジケータをクリックします。ステータスをクリックすることで、2 つの状態が切り替わります。

パブリッシング プラグイン設定の修正

これらの設定は、プラグインの作成者が各自のデータストアの RDU にプラグイン設定を保存するための便利な手段です。パブリッシング プラグイン設定を修正するには、次の手順に従います。

ステップ 1　プライマリ ナビゲーション バーの **Configuration** を選択します。

ステップ 2　セカンダリ ナビゲーション バーの **Publishing** を選択します。Manage Publishing ページが表示されます。

ステップ 3　修正するプラグインに対応するリンクをクリックします。Modify Publishing Plug-Ins ページが表示されます。

Modify Publishing Plug-Ins ページに表示されるフィールドを表 13-11 に示します。

表 13-11　Modify Publishing Plug-ins ページ

フィールド	説明
Plug-In	パブリッシング プラグインの名前を指定します。
Server	データストアがあるサーバの名前を指定します。
Port	データストアがあるポートの番号を指定します。
IP Address	データストアがあるサーバの IP アドレスを指定します。通常、この IP アドレスはサーバ名を使用しない場合に指定します。
User	データストアにアクセスするためのユーザ名を指定します。
Password	データストアにアクセスするためのユーザのパスワードを指定します。
Confirm Password	上で入力したパスワードを確認します。

ステップ 4　Server、Port、IP Address、User、Password、および Confirm Password の各フィールドに必要な値を入力します。これらはすべて必須フィールドなので、これらの情報を入力しなければ、次の操作へ進むことができません。

ステップ 5　**Submit** をクリックして、選択したプラグインに変更を加えます。

自動 FQDN 生成

PacketCable 音声テクノロジーを設定する場合は、音声デバイスごとに完全修飾ドメイン名（FQDN）が BAC データベースに存在する必要があります。これは、KDC でその FQDN の登録サーバにクエリーするからです。BAC の自動 FQDN 生成機能は、1 つの音声テクノロジーに限定されず、すべての BAC テクノロジーで使用可能です。

自動生成の FQDN 形式

BAC は、デバイスの MAC アドレスを使用して、または IPv6 デバイスの DHCP Unique Identifier（DUID）を使用して、FQDN を自動生成します。

MAC アドレスを使用して自動生成される FQDN の形式は次のとおりです。

prefix{*htype-hlen*-aa-bb-cc-dd-ee-ff | 00:00:00:00:00:00:00:00}*suffix.domain*

- *prefix*、*suffix*、および *domain*：BAC 管理者のユーザ インターフェイスまたはプロビジョニング API から設定した情報を指定します。
- *htype*、*hlen*、および aa-bb-cc-dd-ee-ff：デバイスの MAC アドレスを指定します。たとえば、1,6,aa-bb-cc-dd-ee-ff のようになります。
- 00:00:00:00:00:00:00:00：Ipv6 デバイスの DUID を指定します。

prefix と suffix のプロパティは任意指定です。これらのプロパティを指定せず、PacketCable MTA プロビジョニング中にホスト名が指定されない場合、また、prefix と suffix のプロパティがどちらも BAC プロパティ階層で定義されていない場合は、生成される FQDN として、デバイスの MAC アドレスまたはデバイスの DUID にドメイン名が続く形式が使用されます。

FQDN 形式は、次の情報を指定する場合にのみ変更されます。

- プレフィックスおよびデバイス ID：
  prefix{*htype-hlen*-aa-bb-cc-dd-ee-ff | 00:00:00:00:00:00:00:00}.*domain*
- サフィックスおよびデバイス ID：
  {*htype-hlen*-aa-bb-cc-dd-ee-ff | 00:00:00:00:00:00:00:00}suffix.*domain*

次に例を示します。

- プレフィックスが **aaa**、サフィックスが **bbb**、および MAC アドレスが **1,6,aa:bb:cc:dd:ee:ff** のデバイスの場合は、次の FQDN が生成されます。
  aaa1-6-aa-bb-cc-dd-ee-ffbbb.domain
- MAC アドレス（**1,6,aa:bb:cc:dd:ee:ff**）のみのデバイスの場合は、次の FQDN が生成されます。
  1-6-aa-bb-cc-dd-ee-ff.domain
- プレフィックスが **aaa**、サフィックスが **bbb**、および DUID が **00:00:00:00:00:00:00:00** のデバイスの場合は、次の FQDN が生成されます。
  aaa00-00-00-00-00-00-00-00bbb.domain
- DUID **00:00:00:00:00:00:00:00** のみのデバイスの場合は、次の FQDN が生成されます。
  00-00-00-00-00-00-00-00-00-aa.domain
- プレフィックスが **aaa** で MAC アドレスが **1,6,aa:bb:cc:dd:ee:ff** のデバイスの場合は、次の FQDN が生成されます。
  aaa1-6-aa-bb-cc-dd-ee-ff.domain

- サフィックスが **bbb** で MAC アドレスが **1,6,aa:bb:cc:dd:ee:ff** のデバイスの場合は、次の FQDN が生成されます。

  ```
  1-6-aa-bb-cc-dd-ee-ffbbb.domain
  ```

PacketCable および他のテクノロジー用に設定する場合は、ドメイン名プロパティも設定する必要があります。PacketCable MTA のプロビジョニング中にドメイン名を指定しない場合は、BAC プロパティ階層が検索されます。その階層が見つからない場合は、MTA がプロビジョニングされません。

MTA のプロビジョニング中にドメイン名を指定する場合は、BAC プロパティ階層で指定されているドメイン名プロパティに関係なく、そのドメイン名が使用されます。

自動生成 FQDN のプロパティ

プロパティは、BAC プロパティ階層内の任意の許容ポイントで定義できます。システム デフォルト、テクノロジー デフォルト、DHCP 基準、またはサービス クラスを使用して定義を行うことができ、デバイス レベルで定義することもできます。

FQDN 検証

FQDN の生成で使用される情報を入力する場合は、いくつかの点を考慮する必要があります。次の考慮点があります。

- 生成される FQDN で有効な英数字のみを使用する。
- 各ラベルの長さ（生成される FQDN のドット間の文字）を 63 文字未満にする。
- 生成される FQDN の全長が 254 文字を超えないようにする。

（注） FQDN は、RFC1035 によるホスト名とドメイン名をサポートします。

自動 FQDN 生成のサンプル

この項では、自動生成 FQDN の作成例を示します。

ステップ 1　適切なサービス クラスを選択し、このサービス クラスを使用して、*/fqdn/domain* プロパティの値をすべてのデバイスの DNS ドメインに設定します。この例では、example.com というドメインを使用し、ひとまとまりの PacketCable デバイスをそのドメインにプロビジョニングすると想定しています。

（注） ドメインを指定しない場合、サービス クラスのデバイスは BAC から DHCP 設定を受信しません。

ステップ 2　**Submit** をクリックします。

この例では、MAC アドレス 1,6,aa:bb:cc:dd:ee:ff のデバイスにより、1-6-aa-bb-cc-dd-ee-ff.example.com という FQDN が自動生成されます。

また、デバイスのデフォルト設定で、Automatic FQDN Generation オプション ボタンをイネーブルにします。P.13-7 の「デフォルトの設定」を参照してください。

CHAPTER 14

サポートするツールと高度な概念

この章では、Broadband Access Center（BAC）の保守、および製品のインストール、配備、使用の高速化と改善に役立つツールとその使用方法について説明します。

この章では、次のトピックについて説明します。

（注） この項では、ツールの使用方法の例をいくつか示します。多くの場合、ツールのファイル名には *BPR_HOME* と指定されたパスが含まれます。これは、デフォルトのホーム ディレクトリ位置を示しています。

BAC ツール

BAC には、特定の機能をより効率的に実行するための自動ツールが用意されています。表 14-1 に、この BAC リリースでサポートされている各種ツールを示します。

表 14-1　BAC ツール

ツール	説明	参照先
設定ファイル　ユーティリティ	BAC のテンプレートと設定ファイルをテスト、検証、および表示するために使用されます。	設定ファイル ユーティリティの使用方法（P.5-23）
BAC プロセス ウォッチドッグ	BAC ウォッチドッグ デーモンと連動して BAC システム コンポーネントの状態を監視し、サーバを停止または起動します。	コマンドラインからの BAC プロセス ウォッチドッグの使用方法（P.9-2）
RDU ログ レベル ツール	RDU のログ レベルを設定し、デバッグ ログ出力をイネーブルまたはディセーブルにします。	RDU ログ レベル ツールの使用方法（P.10-5）
PacketCable 証明書ツール	KDC で動作するために必要とされる KDC 証明書をインストールおよび管理します。	PKCert.sh ツールの使用方法（P.14-3）
KeyGen ツール	PacketCable サービス キーを生成します。	KeyGen ツールの使用方法（P.14-9）
Network Registrar のプロパティ変更ツール	Cisco Network Registrar DHCP サーバに組み込まれる BAC 拡張が使用するキー設定プロパティを変更するために使用されます。	changeNRProperties.sh ツールの使用方法（P.14-11）
SNMP エージェント設定ツール	SNMP エージェントを管理します。	snmpAgentCfgUtil.sh ツールの使用方法（P.10-11）
診断ツール	システム パフォーマンスおよびトラブルシューティングに関連するサーバ データを収集します。	診断ツールによるトラブルシューティング（P.16-6）
BundleState.sh ツール	サポート拡大のサーバ状態に関連した診断データを組み込みます。	サポートを受けるためのサーバ状態のバンドル（P.16-11）
ディスク容量監視ツール	1 つまたは複数のファイル システムのしきい値を設定します。これらのしきい値を超えると、追加のディスク領域が利用可能になるまでアラートが生成されます。	disk_monitor.sh ツールの使用方法（P.14-13）

PKCert.sh ツールの使用方法

PKCert ツールにより、KDC 証明書とそれに対応する秘密鍵を作成します。また、証明書チェーンを検証し、コピーして、名前を KDC で要求される名前に変更することもできます。

（注） このツールは、KDC コンポーネントがインストールされている場合にのみ使用可能です。

PKCert ツールの実行

PKCert ツールを動作させるには PKCert.sh コマンドを実行します。このコマンドは、デフォルトで *BPR_HOME/kdc* ディレクトリにあります。

シンタックスの説明

PKCert.sh *function option*

- *function*：実行する関数を指定します。次のオプションを選択できます。
  - ***-c***：KDC 証明書を作成します。P.14-3 の「KDC 証明書の作成」を参照してください。
  - ***-v***：PacketCable 証明書セットを検証および正規化します。P.14-4 の「KDC 証明書の検証」を参照してください。
  - ***-z***：*pkcert.log* ファイルに保存されるデバッグ出力のログ レベルを設定します。P.14-5 の「デバッグ出力のログ レベルの設定」を参照してください。

  （注） これらのオプションの使用方法が不明な場合は、***-?*** と指定してヘルプ情報を表示できます。

- *option*：選択する関数に応じて、オプションの関数を実装します。

KDC 証明書の作成

KDC 証明書を作成するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/kdc* にディレクトリを変更します。

ステップ 2　次の構文を使用して PKCert.sh ツールを実行します。

PKCert.sh -s *dir* **-d** *dir* **-c** *cert* **-e -r** *realm* **-a** *name* **-k** *keyFile* [**-n** *serial#*] [**-o**]

- **-s** *dir*：作成元ディレクトリを指定します
- **-d** *dir*：作成先ディレクトリを指定します
- **-c** *cert*：サービス プロバイダーの証明書（DER 暗号化）を使用します
- **-e**：証明書を Euro-PacketCable 証明書として指定します
- **-r** *realm*：KDC 証明書の Kerberos レルムを指定します
- **-a** *name*：KDC の DNS 名を指定します
- **-k** *keyFile*：サービス プロバイダーの秘密鍵（DER 暗号化）を使用します
- **-n** *serial#*：証明書のシリアル番号を設定します
- **-o**：既存ファイルに上書きします

新しい証明書が作成およびインストールされると、その新しい証明書により、サブジェクトの代替名フィールドのレルムが指定されます。新しい証明書は現在の環境に対して固有であり、その環境には次の要素が含まれます。

- KDC レルム。
- マルチメディア ターミナル アダプタ（MTA）が使用する KDC に関連付けられた DNS 名。

例

```
# ./PKCert.sh  -c "-s . -d /opt/CSCObac/kdc/solaris/packetcable/certificates
-k CLCerts/Test_LSCA_privkey.der -c CLCerts/Test_LSCA.cer -r PCTEST.CISCO.COM -n 100
-a kdc.pctest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: .
Destination Directory: /opt/CSCObac/kdc/solaris/packetcable/certificates
Private Key File: CLCerts/Test_LSCA_privkey.der
Certificate File: CLCerts/Test_LSCA.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.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: /opt/CSCObac/kdc/solaris/packetcable/certificates/KDC_private_key.pkcs8
File written:
/opt/CSCObac/kdc/solaris/packetcable/certificates/KDC_private_key_proprietary.
File written: /opt/CSCObac/kdc/solaris/packetcable/certificates/KDC_PublicKey.der
File written: /opt/CSCObac/kdc/solaris/packetcable/certificates/KDC.cer
KDC Certificate Successfully Created at
/opt/CSCObac/kdc/solaris/packetcable/certificates/KDC.cer
```

このコマンドを使用すると、*/opt/CSCObac/kdc/solaris/packetcable/certificates/KDC.cer* ファイルと */opt/CSCObac/kdc/solaris/packetcable/certificates/KDC_private_key.pkcs8* ファイルが作成されます。この KDC 証明書では、レルムが PCTEST.CISCO.COM に、シリアル番号が 100 に、KDC サーバの Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）が kdc.pctest.cisco.com に設定されます。

KDC 証明書の検証

このコマンドにより、指定された作成元ディレクトリのすべてのファイルを調べ、X.509 証明書としての識別を試みます。正規の X.509 証明書が見つかると、ファイルの名前が適正に変更され、作成先ディレクトリにコピーされます。特定の目的（サービス プロバイダーまたはデバイス）での正規の証明書チェーンが複数特定されると、エラーが生成されます。この場合は、作成元ディレクトリから余分な証明書を削除し、もう一度コマンドを実行する必要があります。

（注） **PKCert.sh -v -?** コマンドを入力すると、PKCert ツールを使用した KDC 証明書の検証方法の指示が表示されます。

KDC 証明書を検証するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/kdc* にディレクトリを変更します。

ステップ 2　次の構文を使用して PKCert.sh ツールを実行します。

PKCert.sh -v -s *dir* **-d** *dir* **-r** *dir*

- **-s** *dir*：作成元ディレクトリを指定します
- **-d** *dir*：作成先ディレクトリを指定します
- **-o**：既存のすべてのファイルに上書きします
- **-r** *dir*：参照証明書ディレクトリを指定します

検証は、このパッケージに組み込まれた参照証明書に対して実行されます。**-d** オプションを指定する場合、証明書は名前を正規化して作成先ディレクトリにインストールされます。

例

```
# ./PKCert.sh  -v "-s /opt/CSCObac/kdc/TestCerts -d
/opt/CSCObac/kdc/solaris/packetcable/certificates -o"
Pkcert Version 1.0
Logging to pkcert.log
Output files will overwrite existing files in destination directory

Cert Chain(0)    Chain Type: Service Provider
[Local File]                          [Certificate Label]
[PacketCable Name]
CableLabs_Service_Provider_Root.cer   CableLabs_Service_Provider_Root.cer
Service_Provider.cer                  Service_Provider.cer
Local_System.cer                      Local_System.cer
KDC.cer                               KDC.cer

Cert Chain(1)    Chain Type: Device
[Local File]                          [Certificate Label]
[PacketCable Name]
MTA_Root.cer                          MTA_Root.cer
File written:
/opt/CSCObac/kdc/solaris/packetcable/certificates/CableLabs_Service_Provider_Root.cer
File written: /opt/CSCObac/kdc/solaris/packetcable/certificates/Service_Provider.cer
File written: /opt/CSCObac/kdc/solaris/packetcable/certificates/Local_System.cer
File written: /opt/CSCObac/kdc/solaris/packetcable/certificates/KDC.cer

Service Provider Certificate Chain Written to Destination Directory
/opt/CSCObac/kdc/solaris/packetcable/certificates


File written: /opt/CSCObac/kdc/solaris/packetcable/certificates/MTA_Root.cer

Device Certificate Chain Written to Destination Directory
/opt/CSCObac/kdc/solaris/packetcable/certificates
```

デバッグ出力のログ レベルの設定

このコマンドにより、*BPR_HOME/kdc* ディレクトリの *pkcert.log* に記録されるデバッグ出力のログ レベルを設定できます。ログ ファイルのデータを使用して、要求されたタスクの実行中に発生した問題のトラブルシューティングを行うことができます。

デバッグ出力のログ レベルを設定するには、次の手順に従います。

ステップ 1　*/opt/CSCObac/kdc* にディレクトリを変更します。

ステップ 2　次の構文を使用して PKCert.sh ツールを実行します。

PKCert.sh -s *dir* **-d** *dir* **-k** *keyFile* **-c** *cert* **-r** *realm* **-a** *name* **-n** *serial#* **-o {-z error | info | debug}**

- **-s** *dir*：作成元ディレクトリを指定します

- **-d** *dir*：作成先ディレクトリを指定します
- **-k** *keyFile*：サービス プロバイダーの秘密鍵（DER 暗号化）を使用します
- **-c** *cert*：サービス プロバイダーの証明書（DER 暗号化）を使用します
- **-r** *realm*：KDC 証明書の Kerberos レルムを指定します
- **-a** *name*：KDC の DNS 名を指定します
- **-n** *serial#*：証明書のシリアル番号を設定します
- **-o**：既存ファイルに上書きします
- **-z**：*pkcert.log* ファイルに保存されるデバッグ出力のログ レベルを設定します。次の値を選択できます。
  - **error**：エラー メッセージのロギングを指定します。
  - **info**：情報メッセージのロギングを指定します。
  - **debug**：デバッグ メッセージのロギングを指定します。これはデフォルト設定です。

例

例 1

この例では、ログ レベルがエラー メッセージの収集に設定されています。

```
# ./PKCert.sh  -c "-s /var/certsInput -d /var/certsOutput -k
/var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer -r
PCTEST.CISCO.COM -n 100 -a kdc.pctest.cisco.com -o -z error"
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: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
Setting debug to error
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/solaris/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
```

例 2

この例では、ログ レベルが情報メッセージの収集に設定されています。

```
# ./PKCert.sh -c "-s /var/certsInput
> -d /var/certsOutput
> -k /var/certsInput/Local_System.der
> -c /var/certsInput/Local_System.cer
> -r PCTEST.CISCO.COM
> -n 100
> -a kdc.pctest.cisco.com
> -o -z info"
 INFO [main] 2007-05-02 06:32:26,280 (PKCert.java:97) - Pkcert Version 1.0
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: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
Setting debug to info
 INFO [main] 2007-05-02 06:32:26,289 (PKCCreate.java:69) - PKCCreate startup
WARNING - Certificate File will be overwritten
 INFO [main] 2007-05-02 06:32:26,291 (PKCCreate.java:341) - 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/solaris/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObac/kdc/solaris)
```

例 3

この例では、ログ レベルがデバッグに設定されています。

（注） サンプル出力は、説明のために省略されています。

```
# ./PKCert.sh -c "-s /var/certsInput -d /var/certsOutput -k
/var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer -r
PCTEST.CISCO.COM -n 100 -a kdc.pctest.cisco.com -o -z debug"
 INFO [main] 2007-05-02 06:32:06,029 (PKCert.java:97) - Pkcert Version 1.0
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: IPFONIX.COM
Serial Number: 100
DNS Name of KDC: bacdev3-dpe-4.cisco.com
Setting debug to debug
 INFO [main] 2007-05-02 06:32:06,038 (PKCCreate.java:69) - PKCCreate startup
WARNING - Certificate File will be overwritten
```

```
 INFO [main] 2007-05-02 06:32:06,039 (PKCCreate.java:341) - WARNING - Certificate File
will be overwritten
DEBUG [main] 2007-05-02 06:32:06,054 (PKCert.java:553) - Characters Read: 1218
DEBUG [main] 2007-05-02 06:32:06,056 (PKCert.java:583) - Binary File:
/var/certsInput/Local_System.der Read. Length: 1218
DEBUG [main] 2007-05-02 06:32:06,062 (PKCert.java:553) - Characters Read: 943
DEBUG [main] 2007-05-02 06:32:06,063 (PKCert.java:583) - Binary File:
/var/certsInput/Local_System.cer Read. Length: 943
DEBUG [main] 2007-05-02 06:32:06,064 (PKCert.java:455) - Jar File Path:
/opt/CSCObac/lib/pkcerts.jar
DEBUG [main] 2007-05-02 06:32:06,065 (PKCert.java:456) - Opened jar file:
/opt/CSCObac/lib/pkcerts.jar
DEBUG [main] 2007-05-02 06:32:06,067 (PKCert.java:460) - Jar entry unfiltered:
Tag_Packetcable_Tag/
DEBUG [main] 2007-05-02 06:32:06,068 (PKCert.java:460) - Jar entry unfiltered:
Tag_Packetcable_Tag/CableLabs_Service_Provider_Root.cer
...
DEBUG [main] 2007-05-02 06:32:06,115 (PKCert.java:472) - File:
Tag_Packetcable_Tag/Manu.cer
DEBUG [main] 2007-05-02 06:32:06,116 (PKCert.java:472) - File:
Tag_Packetcable_Tag/Service_Provider.cer
DEBUG [main] 2007-05-02 06:32:06,121 (PKCCreate.java:91) - Found 7 files in jar.
DEBUG [main] 2007-05-02 06:32:06,827 (KDCCert.java:98) - SP Cert subject name:
C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs Local System CA
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01
CableLabs Local System CA
DEBUG [main] 2007-05-02 06:32:07,687 (KDCCert.java:293) - Setting issuer to:
C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs Local System CA
DEBUG [main] 2007-05-02 06:32:07,699 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@bd0b4ea6

DEBUG [main] 2007-05-02 06:32:07,700 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@5035bc0

DEBUG [main] 2007-05-02 06:32:07,701 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@5035bc0

DEBUG [main] 2007-05-02 06:32:07,703 (KDCCert.java:210) - DERCombineTagged [0]
IMPLICIT
    DER ConstructedSequence
        ObjectIdentifier(1.3.6.1.5.2.2)
        Tagged [0]
            DER ConstructedSequence
                Tagged [0]
                    org.bouncycastle.asn1.DERGeneralString@5035bc0
                Tagged [1]
                    DER ConstructedSequence
                        Tagged [0]
                            Integer(2)
                        Tagged [1]
                            DER ConstructedSequence
                                org.bouncycastle.asn1.DERGeneralString@bd0b4ea6
                                org.bouncycastle.asn1.DERGeneralString@5035bc0

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

KeyGen ツールの使用方法

KeyGen ツールを使用して PacketCable サービス キーを生成します。サービス キーは、KDC 通信で必要な Triple Data Encryption Standard（triple DES または 3DES）対称キー（共有秘密情報）です。KDC サーバでは、DPE のプロビジョニング FQDN ごとにサービス キーを必要とします。DPE コマンドライン インターフェイス（CLI）から DPE プロビジョニング FQDN に変更を加えるには、対応する変更を KDC サービス キーのファイル名でも行う必要があります。この変更が必要なのは、KDC サービス キーではそのファイル名の一部として DPE プロビジョニング FQDN が使用されるからです。

KeyGen ツールは、*BPR_HOME/kdc* ディレクトリにあり、DPE プロビジョニング FQDN、レルム名、およびパスワードのコマンドライン引数を使用し、サービス キー ファイルを生成します。

（注） このツールを実行する場合は、DPE で（DPE CLI から **service packetCable** *1..1* **registration kdc-service-key** コマンドを使用して）サービス キーを生成するために使用したのと同じパスワードを入力してください。このパスワードの設定方法の詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

KDC サーバは起動時にサービス キーを読み取ります。サービス キーを修正するには、どんな場合でも KDC サーバを再起動する必要があります。

シンタックスの説明

keygen *options fqdn realm password*

- *options* は次のとおりです。
  - ***-?***：この使用方法メッセージを表示してコマンドを終了します。
  - ***-v*** または ***-version***：このツールのバージョンを表示してコマンドを終了します。
  - ***-q*** または ***-quiet***：出力が何も作成されないクワイエット モードを実装します。
  - ***-c*** または ***-cms***：CMS システムのサービス キーを作成します。
- *fqdn*：DPE の FQDN を指定します。必須の入力項目です。
- *realm*：Kerberos レルムを指定します。必須の入力項目です。
- *password*：使用するパスワードを指定します。これも必須のフィールドです。パスワードの長さは 6 ～ 20 文字にする必要があります。

このファイル名構文を使用して、次の 3 つのサービス キー ファイルが KDC キー ディレクトリに書き込まれます。

mtafqdnmap,*fqdn@REALM*

mtaprovsrvr,*fqdn@REALM*

krbtgt,*REALM@REALM*

- *fqdn*：DPE の FQDN を指定します。
- *REALM*：Kerberos レルムを指定します。

サービス キー ファイルには、必ずバージョン フィールド 0x0000 が含まれています。

例

```
# keygen dpe.cisco.com CISCO.COM changeme
```

このコマンドを実行すると、これらの KDC サービス キーが *BPR_HOME/kdc/solaris/keys* ディレクトリに書き込まれます。

```
mtafqdnmap,dpe.cisco.com@CISCO.COM
mtaprovsrvr,dpe.cisco.com@CISCO.COM
krbtgt,CISCO.COM@CISCO.COM
```

KDC を再起動し、新しいキーが認識されるようにします。次の BAC プロセス ウォッチドッグ コマンドを使用して KDC を再起動します。

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

ここでは、CMS サービス キーの生成例を示します。

```
# keygen -c cms-fqdn.com CMS-REALM-NAME changeme
```

このコマンドを実行すると、この CMS サービス キーが *BPR_HOME/kdc/solaris/keys* ディレクトリに書き込まれます。

```
cms,cms-fqdn.com@CMS-REALM-NAME
```

KDC サービス キーの検証

KDC と DPE でサービス キーを生成したら、両方のコンポーネントでサービス キーが一致するかどうかを検証します。

KeyGen ツールでは、**service packetCable** *1..1* **registration kdc-service-key** コマンドによって DPE でサービス キーを生成するために使用したのと同じパスワードを入力する必要があります。DPE でこのパスワードを設定した後は、*BPR_HOME/dpe/conf* ディレクトリにある *dpe.properties* ファイルからサービス キーを表示できます。*/pktcbl/regsvr/KDCServiceKey=* プロパティに対する値を見つけてください。

次に例を示します。

```
# more dpe.properties
/pktcbl/regsvr/KDCServiceKey=2e:d5:ef:e9:5a:4e:d7:06:67:dc:65:ac:bb:89:e3:2c:bb:
71:5f:22:bf:94:cf:2c
```

（注） この例の出力は、説明のために省略されています。

KDC で生成されたサービス キーを表示するには、*BPR_HOME/kdc/solaris/keys* ディレクトリから次のコマンドを実行します。

od -Ax -tx1 mtaprovsrvr,*fqdn@REALM*

- *fqdn*：DPE の FQDN を指定します。
- *REALM*：Kerberos レルムを指定します。

このコマンドによって生成される出力は、*dpe.properties* ファイルの */pktcbl/regsvr/KDCServiceKey=* プロパティの値と一致します。

次に例を示します。

```
# od -Ax -tx1 mtaprovsrvr,dpe.cisco.com@CISCO.COM
0000000 00 00 2e d5 ef e9 5a 4e d7 06 67 dc 65 ac bb 89
0000010 e3 2c bb 71 5f 22 bf 94 cf 2c
000001a
```

ここで示す例では、KDC で生成されたサービス キーが DPE のサービス キーと一致しています。

changeNRProperties.sh ツールの使用方法

BAC インストール プログラムは、Network Registrar DHCP サーバに組み込まれる BAC 拡張によって使用される、設定プロパティの値を確立します。キー設定プロパティを変更するには、*BPR_HOME/cnr_ep/bin* ディレクトリの **changeNRProperties.sh** コマンドを使用します。

パラメータを何も指定せずにスクリプトを呼び出すと、設定可能なプロパティの一覧を示したヘルプ メッセージが表示されます。

このコマンドを実行するには、次の手順に従います。

ステップ 1　*BPR_HOME/cnr_ep/bin* にディレクトリを変更します。

ステップ 2　次の構文を使用して **changeNRProperties.sh** コマンドを実行します。

changeNRProperties.sh *options*

options は次のとおりです。

- **-help**：ヘルプ メッセージを表示します。**-help** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。
- **-ep enabled** | **disabled**：PacketCable プロパティをイネーブルまたはディセーブルにします。プロパティをイネーブルにするには **-ep enabled** と入力し、ディセーブルにするには **-ep disabled** と入力します。
- **-ec enabled | disabled**：CableHome プロパティをイネーブルまたはディセーブルにします。プロパティをイネーブルにするには **-ec enabled** と入力し、ディセーブルにするには **-ec disabled** と入力します。
- **-d**：現在のプロパティを表示します。**-d** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。
- **-s** *secret*：BAC 共有秘密情報を指定します。たとえば、共有秘密情報がワード *secret* である場合は、**-s secret** と入力します。
- **-f** *fqdn*：RDU FQDN を指定します。たとえば、完全修飾ドメイン名として rdu.example.com を使用する場合は、**-f rdu.example.com** と入力します。
- **-p** *port*：使用する RDU ポートを指定します。たとえば、ポート番号 49187 を使用する場合は、**-p 49187** と入力します。
- **-r** *realm*：PacketCable レルムを指定します。たとえば PacketCable レルムが EXAMPLE.COM の場合は、**-r EXAMPLE.COM** と入力します。

（注）　レルムは大文字で入力する必要があります。

- **-g** *prov_group*：プロビジョニング グループを指定します。たとえば group1 というプロビジョニング グループを使用する場合は、**-g group1** と入力します。
- **-t 00** | **01**：PacketCable TGT をオフまたはオンに設定するかどうかを指定します。たとえば、TGT をオフに設定するには **-t 00** と入力し、オンに設定するには **-t 01** と入力します。
- **-a** *ip*：PacketCable プライマリ DHCP サーバのアドレスを指定します。たとえば、プライマリ DHCP サーバの IP アドレスが 10.10.10.2 の場合は、**-a 10.10.10.2** と入力します。
- **-b** *ip*：PacketCable セカンダリ DHCP サーバのアドレスを指定します。たとえば、セカンダリ DHCP サーバの IP アドレスが 10.10.10.4 の場合は、**-b 10.10.10.4** と入力します。必要に応じて、**-b null** と入力して NULL 値に設定することもできます。
- ***-y*** *ip*：PacketCable プライマリ DNS サーバのアドレスを指定します。たとえば、PacketCable のプライマリ DNS サーバの IP アドレスが 10.10.10.6 の場合は、***-y 10.10.10.6*** と入力します。

- ***-z ip***：PacketCable セカンダリ DNS サーバのアドレスを指定します。たとえば、セカンダリ DNS サーバの IP アドレスが 10.10.10.8 の場合は、***-z 10.10.10.8*** と入力します。必要に応じて、**-z null** と入力して NULL 値に設定することもできます。
- **-o** *prov_ip man_ip*：指定されたプロビジョニング アドレスによって識別される DPE との通信で使用する管理アドレスを設定します。たとえば、プロビジョニング グループの IP アドレスが 10.10.10.7 の場合は **-o 10.10.10.7 10.14.0.4** と入力します。必要に応じて NULL 値を入力することもでき、たとえば **-o 10.10.10.7 null** のように指定します。

ステップ 3　DHCP サーバを再起動します。

例

NR 拡張プロパティ ツールを使用することによって Network Registrar 拡張を変更する例を次に示します。

```
# /opt/CSCObac/cnr_ep_bin/changeNRProperties.sh -g primary1
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.example.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
PacketCable Enable: enabled
CableLabs client TGT: 01
CableLabs client Realm: EXAMPLE.COM
CableLabs client Primary DHCP Server: 10.10.1.2
CableLabs client Secondary DHCP Server: NOT SET
CableLabs client Primary DNS Server: 10.10.1.2
CableLabs client Secondary DNS Server: NOT SET
```

（注）　変更を有効にするには、NR DHCP サーバを再起動する必要があります。

現在のプロパティを表示する例を次に示します。

```
# /opt/CSCObac/cnr_ep_bin/changeNRProperties.sh -d
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.example.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
PacketCable Enable: enabled
CableLabs client TGT: 01
CableLabs client Realm: EXAMPLE.COM
CableLabs client Primary DHCP Server: 10.10.1.2
CableLabs client Secondary DHCP Server: NOT SET
CableLabs client Primary DNS Server: 10.10.1.2
CableLabs client Secondary DNS Server: NOT SET
```

disk_monitor.sh ツールの使用方法

利用可能なディスク領域を監視することは、重要なシステム管理作業です。多数のカスタム スクリプトまたは市販のツールを使用して、この作業を実行できます。

disk_monitor.sh コマンドは、*BPR_HOME/rdu/samples/tools* ディレクトリにあり、1 つ以上のファイル システムのしきい値を設定します。これらのしきい値を超えると、追加のディスク領域が利用可能になるまで、60 秒ごとに Solaris の syslog 機能によってアラートが生成されます。

（注） 少なくとも、**disk_monitor.sh** スクリプトを使用して *BPR_DATA* および *BPR_DBLOG* ディレクトリを監視することをお勧めします。

シンタックスの説明

disk_monitor.sh *filesystem-directory x [filesystem-directory* x*]*

- *filesystem-directory*：監視するファイル システムのディレクトリを示します。
- *x*：指定したファイル システムに適用するしきい値をパーセントで示します。
- *filesystem-directory**：複数のファイル システムを示します。
- *x**：複数のファイル システムに適用するしきい値をパーセントで指定します。

例

例 1

この例では、*/var/CSCObac* ファイル システムの利用率が 80 パーセントに到達した場合に通知が送信されるように指定しています。

```
# ./disk_monitor.sh /var/CSCObac 80
```

データベース ログのディスク領域の利用率が 80 パーセントに達すると、次のようなアラートが syslog ファイルに送信されます。

```
Dec 7 8:16:06 perf-u80-1 BPR: [ID 702911 local6.warning] File system /var/bpr usage is
81% (threshold is 80%)
```

例 2

この例では、disk_monitor.sh ツールをバックグラウンド プロセスとして実行する方法を示しています。コマンドの終わりにアンパサンド（&）を指定すると、バックグラウンドでのプロセスの実行中に出力がただちに返されます。

```
# ./disk_monitor.sh /var/CSCObac 80 &
1020
```

CHAPTER 15

データベースの管理

この章では、RDU データベースの管理および保守について説明します。RDU データベースは、Broadband Access Center（BAC）の中央データベースです。BAC RDU には、十分なディスク容量を確保すること以外、保守は事実上必要ありません。管理者は、データベースのバックアップ手順と回復手順について理解し、熟知している必要があります。

障害復元力について

RDU データベースでは、アプリケーション障害、システム障害、停電のような予測できない問題などが原因で発生するデータベースの破損を防止するために、*先行書き込みロギング*と呼ばれる手法を使用しています。

先行書き込みロギングでは、データベース ファイルに変更を書き込む前に、すべてのデータベース変更の記述をデータベース ログ ファイルに書き込みます。このメカニズムによって、システム障害の原因になった可能性がある不適切なデータベース書き込みを修復できます。

RDU サーバでは、起動されるたびに自動回復が実行されます。この回復処理時に、データとデータベース ファイルとの同期をとるために、データベース ログ ファイルが使用されます。データベース ログには書き込まれたものの、データベースには書き込まれていなかったデータベース変更は、この自動回復の処理中にデータベースに書き込まれます。

このようにして、先行書き込みロギングによって、RDU サーバの再起動時にデータベースが自動的に修復されるので、RDU サーバの障害時にデータベースが破損しないことが実質的に保証されます。

先行書き込みロギングが正しく機能するには、次の条件が必要です。

- ファイル　システムおよび物理ストレージが、要求時にデータを確実に物理ストレージ内にフラッシュするように設定されている必要があります。たとえば、書き込みキャッシュが揮発性メモリだけで構成されたストレージ システムは、システム障害中にデータが失われるので、適切ではありません。ただし、ディスクアレイの書き込みキャッシュがバッテリー バックアップされていて、システム障害時にもデータが失われないことが保証されている場合は正しく機能します。システムがバッテリーによりバックアップされた書き込みキャッシュを備えていない場合、要求時には、メモリ内のデータ キャッシュが実行される代わりにデータ ディスクがフラッシュされます。
- ファイル システムは、RDU データベースのブロック サイズに合せて、ブロック サイズが 8192 バイトに設定されている必要があります。Solaris では、通常、明示的に調整しない限り、この設定がデフォルトです。

データベース ファイル

RDU データベースでは、ファイルが格納されたパーティションをマウントしたファイル システムを使用して、データがバイナリ ファイルに保存されます。システム障害発生後の回復時間が長くならないようにファイル システムを選択し、設定することがきわめて重要です。

データベース ファイルは RDU の動作にとってきわめて重要です。したがって、誤って削除するなどの手作業による操作が行われないように、データベース ファイルを保護するための特別な予防措置を講じる必要があります。これらの重要ファイルを保護するための標準的なシステム管理方法に従ってください。たとえば、これらのファイルには、root ユーザだけがアクセス可能なアクセス権を与える必要があります。また、運用中のシステムには root ユーザとしてログインしないのが最善です。代わりに、root ユーザよりも特権レベルの低いユーザとしてログインし、**sudo** コマンドを使用して root 特権が必要なタスクを実行します。

データベース ストレージ ファイル

RDU サーバは、データベース ディレクトリにある *bpr.db* というファイルに自身のデータベースを格納します。このディレクトリは、*BPR_DATA/rdu/db* ディレクトリにあり、コンポーネントのインストール時に *BPR_DATA* パラメータを指定することでこの場所を設定できます。データベースの移動の詳細については、P.15-9 の「データベースの場所の変更」を参照してください。

(注) 通常、データベース　ファイルはランダムにアクセスされます。したがって、最高のデータベース パフォーマンスを得るためには、シーク時間が最も速く、回転アクセス レイテンシが最も小さいディスクを選択する必要があります。

データベースのトランザクション ログ ファイル

RDU サーバでは、データベースのトランザクション ログは 25 MB のファイルに保存され、そのファイルはデータベースのログ ディレクトリに保存されます。このディレクトリは、*BPR_DBLOG* パラメータを指定することでインストール時に設定します。ログ　　ディレクトリは、*BPR_DBLOG/rdu/dblog* ディレクトリにあります。トランザクション ログの新しいディレクトリへの移動の詳細については、P.15-9 の「データベースの場所の変更」を参照してください。

データベースのログ ファイルの名前には、最初に *log.00000001*、次に *log.00000002* というように連番が付けられます。

(注) トランザクション　ログが保存されるディスクは、通常、シーケンシャルにアクセスされるので、データはログ ファイルの最後に追加されます。最高のデータベース パフォーマンスを達成するには、このアクセス パターンを効率的に処理できるディスクを選択する必要があります。システム上で最も高速なディスクにデータベースのトランザクション ログ ディレクトリを置くことをお勧めします。また、1 GB のディスク容量を使用可能にしてください。

自動ログ管理

データベースのトランザクション ログ ファイルは、トランザクション データのデータベースへの書き込みが完了するまで、そのデータを保存しておくために使用されます。その後、トランザクション ログ データは冗長になり、ファイルがシステムから自動的に削除されます。

通常の状況では、データベースのトランザクション ログ ディレクトリ内にあるログ ファイルの数は数個以内にする必要があります。時間の経過とともに古いトランザクション ログはなくなり、新しいトランザクション ログが作成されます。

（注） データベースのトランザクション　ログは、データベースにとって不可欠です。トランザクション ログ ファイルを手動で削除すると、データベースが破損します。

各種データベース ファイル

データベース ディレクトリには、他にもデータベースの動作に不可欠なファイルが保存されています。これらのファイルは、*rdu.db* ファイルとともに *BPR_DATA/rdu/db* ディレクトリにあり、データベース バックアップの一環としてコピーされます。

- DB_VERSION：データベースの物理的および論理的なバージョンを識別するためのもので、RDU によって内部で使用されます。
- *history.log*：ログ ファイルの自動的な削除、バックアップ、回復、復元処理などの不可欠なデータベース管理タスクに関するロギング動作のために使用されます。このログ ファイルは、管理者に有用な履歴情報を提供するだけでなく、RDU のデータベース処理にとっても非常に重要です。

ディスク容量の要件

完全に実装されたデータベースのサイズは、次の多くの要素で決まります。

- RDU が管理するデバイス オブジェクト
- 各オブジェクト上に保存されているカスタム プロパティ

各パーティションで必要なディスク容量の概算値は、次のとおりです。

- *BPR_DATA*（デバイス オブジェクトごとに約 2 ～ 5 KB）
- *BPR_DBLOG*（500 MB 以上）

注意　これらの数値は、単なる指針として示したものなので、通常のシステム監視の必要がなくなるわけではありません。

disk_monitor.sh ツールを使用すると、使用可能なディスク容量を監視したり、管理者に警告したりできます。詳細については、P.14-13 の「disk_monitor.sh ツールの使用方法」を参照してください。

ディスク容量不足の対処方法

RDU サーバでディスク容量が不足すると、syslog ファシリティを通じて syslog アラートが生成され、RDU ログが記録されます。その後、RDU サーバは自動的に再起動を試みます。RDU サーバが再起動を試みたときに out of disk space エラーが再び発生し、もう一度再起動が試みられる場合があります。

RDU サーバは、空きディスク容量が使用可能になるまで、繰り返し再起動しようとします。空き容量がほぼなくなっているディスク上である程度のディスク領域を解放すると、次の再起動で RDU が正常に起動します。

データベースのサイズが増大して現在のディスク パーティションの容量を超えた場合は、そのデータベースを新しいディスクまたはパーティションに移動してください。詳細については、P.15-9 の「データベースの場所の変更」を参照してください。

（注）　ディスク容量の使用率を監視して障害を防止するのが望ましい方法です。詳細については、P.14-13 の「disk_monitor.sh ツールの使用方法」を参照してください。

バックアップと回復

RDU サーバは、サーバを停止したり、サーバの活動を一時停止したりしなくても実行できる、効率性の高いバックアップ処理をサポートしています。データベースのバックアップおよび回復は、次の手順から構成されます。

- バックアップ：稼働中のサーバから RDU データベースのスナップショットをとります。
- 回復：データベース スナップショットを再利用するための準備を行います。
- 復元：回復したデータベース スナップショットを RDU サーバにコピーします。

（注） 移行が完了したら、必要に応じてデータベースの整合性を確認できます。

これらの手順のそれぞれに自動ツールが用意されています。これらのツールは対話モードまたはサイレント モードで使用できますが、これらのツールを使用するには root 特権が必要です。

データベースのバックアップ

バックアップとは、データベース ファイルをバックアップ ディレクトリにコピーする処理です。そのときに、これらのファイルを圧縮し、テープやその他のアーカイブに保存することができます。

RDU データベースのバックアップは、サーバのアクティビティを中断しないでファイルをコピーするだけなので、非常に効率的です。ただし、RDU データベース ディスクにアクセスするため、バックアップを行うと RDU のパフォーマンスが低下する場合があります。その反対も真実です。バックアップ中の RDU アクティビティは、バックアップのパフォーマンスに悪影響を与えます。そのため、バックアップは、RDU の使用率の低い時間帯に行う必要があります。

バックアップのパフォーマンスは、同時に実行されているシステム アクティビティ以外に、基礎となっているディスクおよびファイル システムのパフォーマンスからも影響を受けます。基本的にバックアップ速度は、データベース ファイルをコピー元からコピー先へコピーする速度と同じです。

BPR_HOME/rdu/bin ディレクトリにある **backupDb.sh** ツールを使用して、データベースのバックアップを行います。

- このツールを使用するには、バックアップ ファイルを保存するバックアップ先ディレクトリを指定する必要があります。このディレクトリは、現在のデータベース ファイル サイズの 120% に相当する使用可能ディスク容量があるディスクまたはパーティション上に存在する必要があります。
- 次の例で説明するように、このツールは、ユーザが指定したディレクトリの下に、タイムスタンプ付きのサブディレクトリを自動的に作成し、そこにバックアップを保存します。また、必要であれば、**–nosubdir** というオプションのフラグを使用して、サブディレクトリの自動作成をディセーブルにできます。

また、**backupDb.sh** コマンドは、自動的に経過を画面にレポートし、その動作を *history.log* に記録します。

backupDb.sh ツールを使用するときに、**–help** オプションを使用すると、使用状況情報を取得できます。

例

この例において、*/var/backup* はデータベース バックアップ ファイルの保存先を示します。

```
# backupDb.sh /var/backup

Database backup started
Back up to: /var/backup/rdu-backup-20070316-031028

Copying DB_VERSION. Size: 396 bytes.
DB_VERSION: 100% completed.

Copying bpr.db. Size: 434176 bytes.
bpr.db: 100% completed.

Copying log.0000000001. Size: 469268 bytes.
log.0000000001: 100% completed.

Copying history.log. Size: 574 bytes.
history.log: 100% completed.

Database backup completed
```

この例では、バックアップするすべてのデータベース ファイルは */var/backup/rdu-backup-20070316-031028* というディレクトリに格納されます。最下位のサブディレクトリ（*rdu-backup-20070316-031028*）は自動的に作成され、現在のタイムスタンプが付けられます。

タイムスタンプ付きのサブディレクトリの形式は、*rdu-backup-yyyyMMdd-HHmmss* です。この例では、サブディレクトリは *rdu-backup-20070316-031028* になります。これは、そのディレクトリに 2007 年 3 月 16 日午前 3 時 10 分 28 秒に開始したバックアップが保存されていることを意味しています。

（注） 移行が完了した後は、**verifyDb.sh** ツールを実行してデータベースの整合性を確認できます。検査はオプションのタスクです。リソースを集中的に使用する動作なので、データベースのバックアップを使用するシステムで実行してください。*BPR_HOME/rdu/internal/db/bin* ディレクトリにある **verifyDb.sh** スクリプトを実行します。

データベースの回復

データベースの回復とは、データベースを一貫性のある状態に戻す処理です。バックアップは稼働中の RDU 上で行われるので、データベースはコピー中に変更される場合があります。ただし、データベースのログ ファイルによって、データベースを一貫性のある状態に戻せることが保証されています。

回復は、データベースのスナップショットに対して行われます。つまり、このタスクは、稼働中の RDU サーバ上のデータベースには作用しません。回復タスクは、バックアップの直後か、または、データベースを RDU サーバに復元する前に実行できます。

（注） シスコでは、バックアップを行うたびに、その直後に回復を行うことを推奨しています。この方法をとると、緊急時にバックアップしたデータベースをより速やかに復元できます。

データベースの回復に要する時間は、バックアップの一環としてコピーするデータベースのログファイルの数によって決まり、したがって、バックアップの実行時の RDU のアクティビティ レベルによって決まります。バックアップの実行時に RDU が同時に実行している活動の数が多いほど、バックアップの一環としてコピーする必要があるトランザクション ログ ファイルの数が多くなり、回復に要する時間が長くなります。一般に、データベースの回復の所要時間は、トランザクション ログ ファイルあたり 10 ～ 60 秒です。

BPR_HOME/rdu/bin ディレクトリにある **recoverDb.sh** ツールを使用して、データベースのスナップショットの回復を行います。このツールを使用するときは、バックアップの場所を指定する必要があります。このディレクトリでは、回復も行われます。

recoverDb.sh ツールを使用するときに、**–help** オプションを使用して、ツールの使用状況情報を取得できます。

例

```
# recoverDb.sh /var/backup/rdu-backup-20070316-031028

*********************************************************
*                                                       *
*  Recovery process modifies the backup snapshot of     *
*  the database.  You should never do recovery without  *
*  making a copy of the database and log files you      *
*  are using for recovery.                              *
*                                                       *
*********************************************************

To start recovery please type "yes" and enter: yes


Database recovery started
Recovering in: /var/backup/rdu-backup-20070316-031028
This process may take a few minutes.
Database recovery completed
```

この例では、*/var/backup/rdu-backup-20070316-031028* ディレクトリにあるスナップショットを一貫性のある状態に回復します。回復処理の経過は画面に表示され、そのアクティビティはスナップショット ディレクトリにある *history.log* ファイルに記録されます。

データベースの復元

データベースの復元とは、あらかじめ回復したデータベースのスナップショットを、RDU サーバによって使用されるデータベース上の場所にコピーする処理です。復元できるのは、あらかじめ回復しておいたデータベースだけです。

データベースの復元とは現行の RDU データベースの交換を意味するので、最初に古いデータベースを適切に削除し、アーカイブすることが非常に重要になります。

注意　置換するデータベースは削除しないでください。古いデータベースのコピーを残しておくと、将来のシステム診断が簡単になる場合があります。

BPR_HOME/rdu/bin ディレクトリにある **restoreDb.sh** ツールを使用して、現行の RDU データベースを別のデータベースと置換します。このツールを使用するときは、入力ディレクトリを指定する必要があります。このディレクトリには、RDU サーバに復元するデータベースの、回復したバックアップ スナップショットが含まれている必要があります。

（注）

restoreDb.sh ツールを実行する前に、**/etc/init.d/bprAgent stop rdu** コマンドを実行して RDU サーバを停止する必要があります。また、データベースをバックアップしてから、すべてのファイルを *rdu/db* ディレクトリおよび *rdu/dblog* ディレクトリから削除する必要もあります。

restoreDb.sh ツールを使用するときに、**-help** オプションを使用すると、使用状況情報を取得できます。

例

```
# restoreDb.sh /var/backup/rdu-backup-20070316-031028

Restoring RDU database...
Restoring from: /var/backup/rdu-backup-20070316-031028

Copying bpr.db. Size: 434176 bytes.
bpr.db: 100% completed.

Copying log.0000000001. Size: 471261 bytes.
log.0000000001: 100% completed.

Copying history.log. Size: 1260 bytes.
history.log: 100% completed.

Copying DB_VERSION. Size: 396 bytes.
DB_VERSION: 100% completed.

Database was successfully restored
You can now start RDU server.
```

この例では、*/var/backup/rdu-backup-20070316-031028* ディレクトリにあるデータベースが RDU サーバに復元されます。

復元処理の完了後は、RDU を再起動する必要があります。RDU ログ ファイルには、起動が成功したことを示すメッセージが書き込まれます。

このツールでは、経過がユーザに報告され、動作が *history.log* ファイルに記録されます。

データベースの場所の変更

あるパーティションまたはディスクから、同じシステム上の別のパーティションまたはディスクにデータベースを移動することができます。管理上の理由で、この処理が必要になる場合があります。この処理を行うには、RDU サーバおよび BAC プロセス ウォッチドッグを停止する必要があります。

データベースの場所を変更する処理では、システム パラメータを変更し、該当のファイルを新しい場所にコピーします。

次のパラメータの一方または両方を調整できます。

- *BPR_DATA*：このパラメータは、インストール時に最初に設定され、データベース、およびログや設定ファイルなどその他多くの重要なファイルが保存されるディレクトリを示しています。

  また、このディレクトリには、特に BAC プロセス ウォッチドッグ、DPE（同じシステムにインストールされている場合）、RDU、および SNMP エージェントのログ データが格納されます。
- *BPR_DBLOG*：このパラメータは、インストール時に最初に設定され、データベースのトランザクション ログ ファイルが保存されるディレクトリを示しています。

上記のパラメータの値は、*BPR_DATA/bpr_definitions.sh* というファイルに記録されます。このファイルを変更した場合、システム上で実行されているすべての BAC コンポーネントを再起動する必要があります。

データベースおよびトランザクション ファイルの場所を変更するには、次の手順に従います。

ステップ 1　**/etc/init.d/bprAgent stop** コマンドを実行して BAC プロセス ウォッチドッグおよびすべての BAC コンポーネントを停止します。

ステップ 2　*BPR_HOME/bpr_definitions.sh* ファイルのバックアップ コピーを作成します。

ステップ 3　このファイルを編集して、*BPR_DATA* パラメータおよび *BPR_DBLOG* パラメータの一方または両方を新しいディレクトリに変更します。

ステップ 4　このファイルを保存します。

ステップ 5　*BPR_DATA* ディレクトリ、*BPR_DBLOG* ディレクトリ、またはその両方の元のディレクトリ構造および内容を、新しい場所にコピーまたは移動します。コピーを行う場合は、すべてのファイルおよびディレクトリのアクセス権が維持されるようにしてください。

ステップ 6　**/etc/init.d/bprAgent start** コマンドを実行して、BAC プロセス ウォッチドッグおよびすべての BAC コンポーネントを起動します。

ステップ 7　該当のログ ファイルを監視して、すべてのコンポーネントが正常に起動したことを確認します。

RDU データベースの移行

RDU データベースの移行の詳細については、『*Installation and Setup Guide for Cisco Broadband Access Center 4.0*』を参照してください。

CHAPTER 16

Broadband Access Center のトラブルシューティング

この章では、Broadband Access Center（BAC）のトラブルシューティングを行う方法の詳細について説明します。この章は、次の項で構成されています。

BAC プロビジョニングに関連する FAQ のリストについては、付録 E「Broadband Access Center のプロビジョニングに関する FAQ」を参照してください。

トラブルシューティングのチェックリスト

BAC のトラブルシューティングでは、表 16-1 に示すチェックリストを使用します。

表 16-1　トラブルシューティングのチェックリスト

手順	参照先	確認
1. BAC コンポーネントがインストールされているすべてのシステムで、BAC のプロセスが稼働しているかどうかを確認します。	コマンドラインからの BAC プロセス ウォッチドッグの使用方法（P.9-2）	❑
2. BAC のコンポーネント ログで、重大度の高いエラーが示されていないかどうかを確認します。これには、次のものに関して記録された情報が含まれます。 — RDU — DPE	RDU のログ（P.10-4） DPE のログ（P.10-8）	❑
3. 管理者のユーザ インターフェイスからサーバのアップ タイムを表示し、サーバがバウンスしていないことを確認します。	サーバの表示（P.12-23）	❑
4. 管理者のユーザ インターフェイスから、RDU および DPE のサービス パフォーマンス統計情報を表示します。トランザクション時間が長くなっているなど、異常な数値がないか確認します。	サーバの表示（P.12-23）	❑
5. syslog アラート ログを確認します。	付録 A「アラートとエラー メッセージ」	❑
6. 次のようなオペレーティング システムおよびハードウェアのリソースを確認します。 — ディスク領域 — CPU 時間 — メモリ	特定のコマンドについては、Solaris のマニュアルを参照してください。	❑
7. 特定のデバイスのトラブルシューティングを行う場合は、DPE でキャッシュされているデバイス命令を表示します。	**show device-config** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）	❑
8. 管理者のユーザ インターフェイスから、個々のデバイスのトラブルシューティングを設定します。しばらく経過してから、トラブルシューティング ログを調べます。	トラブルシューティングのためのデバイスの設定（P.16-3）	❑
9. RDU または適切な DPE でより高いロギング レベルを設定し、詳細なログ情報を取得します。	RDU ログ レベル ツールの使用方法（P.10-5） **log level** コマンド（『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照）	❑

デバイス ID に基づくデバイスのトラブルシューティング

この機能を使用すると、1 つ以上の特定のデバイスに関する詳細な診断情報を収集できます。トラブルシューティング情報には、特定のデバイスまたはデバイス グループに関連するサーバ インタラクションがすべて含まれます。この情報には、管理者のユーザ インターフェイスの操作、RDU Application Programming Interface（API; アプリケーション プログラミング インターフェイス）の操作、DPE とデバイスとのインタラクション、およびサーバ間の DPE と RDU のインタラクションも含まれます。

1 つ以上の特定のデバイスに対して、ノード管理によって診断をイネーブルまたはディセーブルにできます。この場合、ロギングをオンにしたり、特定のデバイス情報についてのログ ファイルを検索したりする必要はありません。

BAC はデバイス ID（MAC アドレスと DUID）に基づいてデバイスのリストを保持しており、それについての詳細な診断情報が収集されます。トラブルシューティング情報は RDU で一元的に保管され、デバイス単位で保持されます。DPE と Cisco Network Registrar 拡張は、どちらもこのデータを保存しません。この情報は RDU に転送されます。RDU は、その情報を受信すると、*BPR_DATA/rdu/logs* ディレクトリの *troubleshooting.log* ファイルに書き込みます。

troubleshooting.log ファイルは、その他の *rdu.log*、*dpe.log*、および *audit.log* などのログ ファイルとは異なります。診断モードになっている特定のデバイス セットに関連する詳細なトラブルシューティング情報のみが記録されます。

DPE または Network Registrar 拡張から RDU への接続が失われた場合、DPE または Network Registrar 拡張で発生している新しいトラブルシューティング イベントはすべて廃棄されます。トラブルシューティング情報のロギングが再開されるのは、RDU への接続が復元された場合だけです。

DPE は、診断される特定のデバイスの MAC アドレスと DUID をそのデバイスの IP アドレスにマッピングします。DPE は、診断されるデバイスの Network Registrar 拡張から IP アップデートを受信します。

新しいデバイスやグループの追加など、デバイス トラッキング リストに対するすべての修正は、すべてのサーバでただちに実行されます。RDU または DPE をリブートする必要はありません。各サーバのログ ファイルには、診断モードになっているデバイスの現在のリストが示されます。

注意　デバイスのトラブルシューティング機能を使用する場合は、追加のメモリおよびディスク領域が必要になります。トラッキング対象のデバイス数が増えると、作成されたログの数をサポートするのに必要なメモリおよびディスク領域の容量も増えます。

トラブルシューティングのためのデバイスの設定

デバイス診断は、1 つ以上のデバイスが診断モードに設定されるまでディセーブルになっています。

デバイスの診断をイネーブルにするには、そのデバイスを BAC RDU で事前登録しておく必要があります。デバイスが事前登録されていない場合は、Manage Devices ページで Add ボタンをクリックして、デバイスを追加します。デバイスの追加については、P.12-14 の「デバイス レコードの追加」を参照してください。

診断モードになるデバイスの最大数を設定すると、気付かないうちに膨大な数のデバイスをこのモードに移行して、サーバのパフォーマンスを低下させてしまうことを回避できます。デフォルトでは、この数が 25 に設定されています。管理者のユーザ インターフェイスからトラブルシューティング モードに移行できるデバイスの最大数を設定するには、Systems Defaults ページで **Configuration** > **Defaults** タブの順にクリックします。Maximum Diagnostics Device Count フィールドに値を入力します。

ノードへのデバイスの関連付け

デバイスは、特定のノードに関連させることによってトラブルシューティングを行うことができます。関連付け機能により、MAC アドレスまたは DUID を使用してデバイスを特定のノードに関連付けます。さらにその特定のノードは、特定のノード タイプに関連付けられます。（P.12-17 の「デバイスの関連付けと関連付け解除」を参照してください）。関連付けによってデバイスについての膨大な量の情報が記録されるので、それらの情報に基づいて潜在的な問題のトラブルシューティングを実行できます。

表 16-2 に、関連付け機能と関連付け解除機能を使用したワークフローの例を示します。

表 16-2　関連付け / 関連付け解除プロセスのサンプル

手順	作業
1.	問題が存在するかどうかを判断し、影響を受けるデバイスを識別します。
2.	デバイスをノードに関連付けます。
3.	デバイスのトラフィックが確実に通過するように数分待つか、またはデバイスのハードブートを実行します。
4.	ワード プロセッシング アプリケーションで *BPR_DATA/rdu/logs/troubleshooting.log* ファイルを開き、特定のデバイスの MAC アドレスまたは DUID のエントリを見つけます。
5.	問題を識別、訂正、テスト、および検証します。
6.	デバイスをノードから関連付け解除します。

診断モードになっているデバイスのリストの表示

デバイスのトラブルシューティングをイネーブルにすると、そのデバイスは、トラブルシューティング モードのデバイスのリストを含む、特別なデバイス ノードに自動的に追加されます。ノード タイプは **system** で、ノード名は **system-diagnostics** です。このグループ内のデバイスのリストには、API または管理者のユーザ インターフェイスからアクセスできます。

診断が現在イネーブルになっているデバイスのリストを表示するには、次の手順に従います。

ステップ 1　Manage Devices ページで、Search Type ドロップダウン リストをクリックし、Node Search を選択します。

ステップ 2　Node Name (Node Type) ドロップダウン リストから、診断モードのデバイスすべてを表示するための、**system-diagnostics (system)** オプションを選択します。

ステップ 3　**Search** をクリックします。

（注）　上記のほか、診断モードのデバイスのリストを表示するには、RDU ログ（*rdu.log*）ファイルおよび DPE ログ（*dpe.log*）ファイルを調べるという方法もあります。デバイスのリストの記録は、サーバが起動するたび、および診断がイネーブルになっているデバイスのリストが変更されるたびに行われます。

診断がイネーブルになっているデバイスは、ログ レベルが 5（通知）に設定された状態でログ ファイルに表示されます。ログ ファイルの詳細については、P.10-2 の「イベントのロギング」を参照してください。

例

次の例では、MTA のトラブルシューティングを行う間のログ出力を示しています。

```
bac-test.example.com:2005 03 04 18:38:24 EST:%BAC-DIAGNOSTICS-3-4055:[##MTA-9a
Unconfirmed FQDN Request Received from [/10.10.10.5 ['kdcquery']. Client with IP
Address [10.10.20.2] and MAC Address [1,6,00:00 :ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:24 EST:%BAC-DIAGNOSTICS-3-4082:[Results of BACC
Lookup. FQDN: [1-6-00-00-ca-b7-7e-91.example.com MAC: 1,6,00:00:ca:b7:7e:91. Client
with IP Address [10.10.20.2] and MAC Address [1,6,00:00:ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:24 EST:%BAC-DIAGNOSTICS-3-4070:[##MTA-9b FQDN
Reply Sent to [/10.10.20.2(41142) for MTA 1,6,00:00:ca:b7:7e:91. Client with IP
Address [10.10.20.2] and MAC Address [1,6, 00:00:ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:26 EST:%BAC-DIAGNOSTICS-3-4132:[##MTA-13
Incoming APREQ received from [/10.10.20.2:1293. Client with IP Address [10.10.20.2]
and MAC  Address [1,6,00:00:ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:26 EST:%BAC-DIAGNOSTICS-3-4141:[##MTA-13 APREP
sent to [/10.10.20.2(1293) For MTA 1,6,00:00:ca:b7:7e:91. Client with IP Address
[10.10.20.2] and MAC Address [1,6,00:00: ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:26 EST:%BAC-DIAGNOSTICS-3-0764:[[##MTA-15 SNMPv3
INFORM Received From 10.10.20.2. Client with IP Address [10.10.20.2] and MAC Address
[1,6,00:00:ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:26 EST:%BAC-DIAGNOSTICS-3-0764:[[##MTA-19 SNMPv3
SET Sent to 10.10.20.2. Client with IP Address [10.10.20.2] and MAC Address
[1,6,00:00:ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:26 EST:%BAC-DIAGNOSTICS-3-1092:[Received a TFTP
[read] request from [10.10.20.2:1271] for [bpr01060000cab77e910002]; Client with MAC
Address [1,6,00:00:ca:b7:7e:91] and IP Address [10.10.20.2]]
bac-test.example.com:2005 03 04 18:38:26 EST:%BAC-DIAGNOSTICS-3-1155:[##MTA-23
Finished handling [read] request from [10.10.20.2:1271] for [bpr01060000cab77e910002];
Transferred [236] bytes to Client with MAC Address [1,6,00:00:ca:b7:7e:91] and IP
Address [10.10.20.2]]
bac-test.example.com:2005 03 04 18:38:27 EST:%BAC-DIAGNOSTICS-3-0764:[[##MTA-25 SNMP
Provisioning State INFORM Received from 10.10.20.2. Client with IP Address
[10.10.20.2] and MAC Address [1,6,00:00:ca:b7:7e:91]]]
bac-test.example.com:2005 03 04 18:38:27 EST:%BAC-DIAGNOSTICS-3-0764:[[MTA
Configuration Confirmed, Returned 'pass' as the final MTA provisioning state for
10.10.20.2. Client with IP Address [10.10.20.2] and MAC Address
[1,6,00:00:ca:b7:7e:91]]]
```

診断ツールによるトラブルシューティング

診断ツールを使用すると、BAC サーバのパフォーマンスの統計情報を特定のタイプの統計にまで掘り下げて収集することができます。このツールで実行されるタスクごとに個別のスクリプトを使用すると、次の作業を実行できます。

- 診断情報を同時に収集する（**startDiagnostics.sh**）
- 診断を途中で中止する（**stopDiagnostics.sh**）
- 診断情報の収集ステータスを判断する（**statusDiagnostics.sh**）

診断ツールは、問題が発生したためにトラブルシューティング用の追加データが必要になったときに同時に実行したり、cron ジョブによって指定されたスケジュールで定期的に実行されるように設定したりすることができます。

注意　診断ツールを使用する場合は、診断データを保存するための十分なスペースをシステムで確保してください。

診断ツールは次の場所にあります。

- RDU：*BPR_HOME/rdu/diagnostics/bin*
- DPE：*BPR_HOME/dpe/diagnostics/bin*
- Cisco Network Registrar：*BPR_HOME/cnr_ep/diagnostics/bin*

（注）　収集した診断情報は、**bundleState.sh**　スクリプトを使用してバンドルできます。詳細については、P.16-11 の「サポートを受けるためのサーバ状態のバンドル」を参照してください。

startDiagnostics.sh ツールの使用方法

startDiagnostics.sh ツールは次の 2 種類のモードで実行できます。

- 対話：このモードでは、必要な診断データをオプションのリストから選択できます。
- 非対話：このモードでは、引数が書き込まれた応答ファイルを最初に生成します。次に、**startDiagnostics.sh** スクリプトを実行します。このツールにより、応答ファイルで指定されている引数に基づいて診断データが収集されます。

シンタックスの説明　***startDiagnostics.sh*** [***-r*** *response_file*] | [***-g*** *response_file*] [***-help***]

- ***startDiagnostics.sh***：対話モードで診断を実行します。
- *response_file*：応答ファイルを指定します。
- ***-r*** *response_file*：非対話モードで診断ツールを実行するために生成された応答ファイルを使用します。
- ***-g*** response_file：診断を実行せずに応答ファイルを生成します。
- ***-help***：ツールのヘルプを表示します。***-help*** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。

対話モードでの startDiagnostics.sh の実行

引数を何も指定しないで **startDiagnostics.sh** を入力すると、診断ツールは対話モードで実行され、RDU、DPE、および Network Registrar の各サーバから収集する統計情報を選択するよう求めるメッセージが表示されます。

注意　システム パフォーマンスに深刻な影響を与える可能性があるので、統計情報は慎重に処理してください。

シンタックスの説明　***startDiagnostics.sh*** [***-help***]

- ***startDiagnostics.sh***：対話モードで診断を実行します。
- ***-help***：ツールのヘルプを表示します。***-help*** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。

例

```
# ./startDiagnostics.sh

Please enter directory where to put output files [] /var/CSCObac
   Please enter the duration of the diagnostic (sec) [600]

Please select statistics you would like to gather on RDU

   CPU statistics (y/n/q)? [y]
   Process statistics (y/n/q)? [n]
   IO statistics (y/n/q)? [y]
   Memory statistics (y/n/q)? [y]
   Network statistics (y/n/q)? [y]
   RDU API traffic (y/n/q)? [y]
   RDU CNR traffic (y/n/q)? [y]
   RDU DPE traffic (y/n/q)? [y]
   RDU CNR extension traffic (y/n/q)? [y]
   RDU SNMP traffic (y/n/q)? [y]
   System Configuration (y/n/q)? [y]

Enter addition argument for RDU API traffic
   Please enter RDU Server port [49187]

Enter addition arguments for RDU DPE traffic
   Enter DPE ip addr if you want to capture traffic by ip addr [] 10.10.29.1
   Enter DPE port number if you want to capture traffic by port number [] 49186

Enter addition arguments for RDU CNR_EX traffic
   Enter Ip addr if you want to capture traffic by Cnr Extension IP addr [] 10.10.85.2
   Enter port number if you want to capture traffic by Cnr Extension port []

You could run statusDiagnostics.sh to find out the status of the diagnostics.
You could run stopDiagnostics.sh to stop the diagnostics.
You could run bundleState.sh to bundle the output when diagnostics is complete.
```

（注）　次のオプションの統計をイネーブルにしていない場合、ツールは例にある追加引数の値を要求しません。

- RDU-API トラフィック
- RDU-DPE トラフィック
- RDU-Network Registrar 拡張トラフィック

startDiagnostics.sh ツールを実行すると、ツールを実行したディレクトリの下位に統計ごとの出力ファイルが作成されます。出力ファイルをバンドルし、Cisco Technical Assistance Center に転送してサポートを受けることもできます。サポートを受けるには、System Diagnostics Capture プロンプトで **y** と入力します。

次に例を示します。

```
System Configuration (y/n/q)? [y]
```

サーバ状態のバンドルの詳細については、P.16-11 の「サポートを受けるためのサーバ状態のバンドル」を参照してください。

非対話モードでの startDiagnostics.sh の実行

非対話モードで **startDiagnostics.sh** ツールを初めて実行する前に、応答ファイルを生成する必要があります。その後、1 つのコマンドだけを実行すると、応答ファイルにある引数に基づいて診断情報が収集されます。

シンタックスの説明

startDiagnostics.sh {***-g*** *response_file* | ***-r*** *response_file*} [***-help***]

- ***-g***：応答ファイルを生成します。このオプションは、応答ファイルを初めて生成する場合にのみ使用する必要があります
- ***-r***：応答ファイルを使用して診断ツールを実行します。
- *response_file*：応答ファイルの名前を指定します。
- ***-help***：ツールのヘルプを表示します。***-help*** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。

例

応答ファイルを生成するときの結果を次に示します。

```
# ./startDiagnostics.sh -g response.txt

Please enter directory where to put output files [] /var/CSCObac
   Please enter the duration of the diagnostic (sec) [600]

Please select statistics you would like to gather on RDU

   CPU statistics (y/n/q)? [y]
   Process statistics (y/n/q)? [n]
   IO statistics (y/n/q)? [y]
   Memory statistics (y/n/q)? [y]
   Network statistics (y/n/q)? [y]
   RDU API traffic (y/n/q)? [y] n
   RDU CNR traffic (y/n/q)? [y]
   RDU DPE traffic (y/n/q)? [y] n
   RDU CNR extension traffic (y/n/q)? [y] n
   RDU SNMP traffic (y/n/q)? [y]
   System Configuration (y/n/q)? [y]

Finished generate response file (response.txt).
```

response.txt は、**startDiagnostics.sh** スクリプトを実行するディレクトリの下位ディレクトリに生成されます。この場合は、*BPR_HOME/rdu/diagnostics/bin* です。RDU 診断用に生成される応答ファイルのサンプルを次に示します。

```
test.bundle.direcotry=/var/CSCObac
test.bundle.duration.sec=100
test.cpu.enable=true
test.process.enable=false
test.io.enable=true
test.memory.enable=true
test.network.enable=true
test.rdu_api_traffic.enable=true
test.rdu_cnr_traffic.enable=true
test.rdu_dpe_traffic.enable=true
test.rdu_cnr_ex_traffic.enable=true
test.rdu_snmp_traffic.enable=true
test.system_config.enable=true
test.rdu.port=49187
test.dpe.port=49186
test.dpe.ip=10.10.29.1
test.cnr_ex.ip=10.10.85.2
test.cnr_ex.port=
EOF
```

生成した応答ファイルを使用して診断ツールを実行したときの結果を次に示します。

```
# ./startDiagnostics.sh -r response.txt

You could run statusDiagnostics.sh to find out the status of the diagnostics.
You could run stopDiagnostics.sh to stop the diagnostics.
```

startDiagnostics.sh ツールを実行すると、ツールを実行したディレクトリの下位に統計ごとの出力ファイルが作成されます。

statusDiagnostics.sh ツールの使用方法

statusDiagnostics.sh ツールを使用して、必要な統計情報の診断収集のステータスを判断します。

シンタックスの説明

statusDiagnostics.sh により、統計情報ごとに診断収集のステータスを表示します。

（注） ***statusDiagnostics.sh*** ツールでは ***-help*** オプションを使用できません。

例

```
# ./statusDiagnostics.sh
   CPU diagnostic is running.
   Process diagnostics stopped.
   IO diagnostic is running.
   Memory diagnostic is running.
   Network diagnostic is running.
   Rdu api traffic diagnostic is running.
   Rdu cnr traffic diagnostic is running.
   Rdu dpe traffic diagnostic is running.
   Rdu cnr_ex traffic diagnostic is running.
   Rdu snmp traffic diagnostic is running.
```

stopDiagnostics.sh ツールの使用方法

stopDiagnostics.sh ツールを使用して、統計情報の 1 つまたはすべてに対する診断の実行を中止します。このツールは、対話モードまたは非対話モードで実行できます。

対話モードでの stopDiagnostics.sh の実行

何も引数を指定せずに **stopDiagnostics.sh** を対話モードで実行すると、すべての統計情報または特定の統計情報の診断を中止するかどうかを尋ねるメッセージが表示されます。

シンタックスの説明　*stopDiagnostics.sh* [*-help*]

- ***stopDiagnostics.sh***：対話モードでの診断収集を中止します。
- ***-help***：ツールのヘルプを表示します。***-help*** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。

例

```
# ./stopDiagnostics.sh

This script allowed to stop specific diagnostic or all diagnostics.
If you would like to stop specific diagnostics, say no to question below.

   Would you like to stop all diagnostics (y/n/q)? [y]
```

非対話モードでの stopDiagnostics.sh の実行

stopDiagnostics.sh を非対話モードで実行すると、すべての統計の診断が中止されます。

シンタックスの説明　*stopDiagnostics.sh -a* [*-help*]

- ***-a***：メッセージが表示されることなく、すべての統計に対する診断が中止されます。
- ***-help***：ツールのヘルプを表示します。***-help*** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。

例

```
# ./stopDiagnostics.sh -a
#
```

サポートを受けるためのサーバ状態のバンドル

BPR_HOME/{rdu | dpe}/diagnostics/bin ディレクトリにある診断ツールを使用して、サーバ設定や他の診断情報を生成できます（これらのツールの実行方法については、P.16-6 の「診断ツールによるトラブルシューティング」を参照してください）。サポートを受けるためにこの診断情報を Cisco Technical Assistance Center に送信するには、診断ツールを使用して作成される出力ディレクトリをバンドルしてアーカイブを作成する必要があります。このタスクを実行するには、**bundleState.sh** ツールを使用します。

bundleState.sh ツールによって診断情報が収集されるわけではありません。**startDiagnostics.sh** などのツールによって収集されるデータの zip ファイルと tar ファイルを作成するだけです。

バンドルする診断情報には、少なくともシステム設定に関連した情報を含める必要があります。システム情報を生成するには、次のいずれかのツールを使用します。

- captureConfiguration.sh：マウントとディスクの設定、メモリ、およびオペレーティング システムとハードウェアのデータなどのシステム設定情報を収集します。このスクリプトを実行する場合は、出力ディレクトリを指定する必要があります。
- startDiagnostics.sh：BAC サーバのパフォーマンス統計情報を収集します。このスクリプトを実行してシステム設定を取り込む場合は、System Configuration プロンプトで **y** と入力する必要があります。次に例を示します。

  ```
  System Configuration (y/n/q)? [y]
  ```

  詳細については、P.16-6 の「startDiagnostics.sh ツールの使用方法」を参照してください。

問題によっては、追加の診断情報を収集してバンドルに追加するようシスコのサポート担当者から指示される場合があります。

シンタックスの説明

bundleState.sh *archive_directory output_directory* [***-help***]

- *archive_directory*：バンドルするディレクトリ。
- *output_directory*：バンドルの出力先ディレクトリ。
- ***-help***：ツールのヘルプを表示します。***-help*** オプションは排他的に使用する必要があります。他のオプションと一緒に使用しないでください。

例

```
# ./bundleState.sh /var/CSCObac /var/CSCObac
/var/CSCObac/state-20071129-064042
Creating state bundle for Cisco support...
+ /var/CSCObac/state-20071129-064042.bpr
+ Compressing state bundle...
+ Size: 3736K compressed, 83776K uncompressed
```

DOCSIS ネットワークのトラブルシューティング

BAC および Cisco uBR7246 CMTS に関する DOCSIS テクノロジーのトラブルシューティングの詳細については、次のアドレスにある『*Troubleshooting uBR Cable Modems Not Coming Online*』を参照してください。
http://www.cisco.com/en/US/tech/tk86/tk89/technologies_tech_note09186a0080094eb1.shtml

PacketCable eMTA プロビジョニングのトラブルシューティング

この項では、PacketCable 音声テクノロジーの配備において考えられる問題の解決に役立つ情報を提供します。

- トラブルシューティングのツール（P.16-15）
- トラブルシューティングのシナリオ（P.16-16）
- 証明書信頼階層（P.16-20）

この項では、PacketCable Multimedia Terminal Adapter（MTA; マルチメディア ターミナル アダプタ）デバイスのプロビジョニング仕様（PKT-SPPROV1.5-I01-050128）の内容を理解していることを前提としています。詳細については、PacketCable の Web サイトを参照してください。

プロビジョニング PacketCable 組み込み型 MTA（eMTA）は、比較的複雑なプロセスですが、適切なツールを使用し、要領を理解すれば、簡単に eMTA を使用することができます。

この項では、Network Registrar と BAC の両方が使用中であることを前提としていますが、情報の多くは他の配備環境にも当てはまります。Network Registrar の基礎知識（スコープ、ポリシー、基本的な DNS ゾーン設定、およびレコード エントリ）および BAC の基礎知識（サービス クラス、DHCP 基準、ファイル、および BAC ディレクトリ構造）があることを前提としています。

PacketCable eMTA プロビジョニング プロセスは、セキュアなフローを実現するために 25 のステップで構成されています。基本フローの手順数はそれよりも大幅に少ない数です。eMTA のトラブルシューティングを行うには、PacketCable プロビジョニング仕様にある 25 のステップについての知識が不可欠です。第 7 章「PacketCable 音声設定」を参照してください。

この項では、次のトピックについて説明します。

- コンポーネント（P.16-12）
- 主要な変数（P.16-14）

コンポーネント

eMTA のトラブルシューティングを行う前に、次のシステム コンポーネントを理解してください。

- eMTA
- DHCP サーバ
- DNS サーバ
- KDC
- PacketCable プロビジョニング サーバ
- コール管理サーバ

eMTA

eMTA はケーブル モデムと MTA で構成され、共通のソフトウェア イメージを備えており、1 つのボックスに組み込まれています。CM と MTA はそれぞれ独自の MAC アドレスを持ち、それぞれが DHCP を実行して固有の IP アドレスを取得します。eMTA には、最低でも 3 つの証明書があります。1 つは固有の MTA 証明書です。2 つ目の証明書は MTA の製造業者を特定します。デバイスと製造業者の証明書は、両方とも認証で使用するために MTA によって KDC に送信されます。3 つ目の証明書は、KDC から MTA に送信される証明書を検証するために使用されるテレフォニー ルート証明書です。KDC 証明書はテレフォニー ルートをルートとする証明書チェーンに組み込まれるため、そのテレフォニー ルートは、KDC 証明書の正当性を検証するために MTA に存在する必要があります。MTA 部分では独自の設定ファイルを受信し、制御するコール エージェントを特定するために使用します。

DHCP サーバ

DOCSIS 仕様では、DHCP を使用してケーブル モデムがその IP アドレスをネゴシエートするように規定しています。MTA は、DOCSIS ネットワークのほとんどの CPE と同様に、DHCP を使用して IP アドレスや他の重要な情報（DNS サーバ、Kerberos レルム名の PacketCable Option 122、プロビジョニング サーバの FQDN）を取得する必要があります。

（注） ケーブル モデム部分では、通常必要とされる DHCP オプションの他に、Option 122 のサブオプション 1 を要求して受信する必要があります。ケーブル モデム部分は、オファーを受信するときの正しい送信元 DHCP サーバの IP アドレスとして、そのサブオプションを MTA 部分に渡します。

PacketCable サポート付きの BAC を使用する場合は、BAC の設定が正しければ、ToD サーバ、DNS サーバ、TFTP サーバ、および Option 122 のフィールドに値が自動的に取り込まれます。これらのフィールドを Network Registrar ポリシーで明示的に設定する必要はありません。

DNS サーバ

Domain Name System（DNS; ドメイン ネーム システム）サーバは、PacketCable プロビジョニングの基本的な要素です。PacketCable プロビジョニング サーバは、BAC アーキテクチャでの Device Provisioning Engine（DPE）です。Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）が DHCP サーバにより Option 122 で MTA に提供されるため、適切なゾーンのアドレス（A）レコードを持っている必要があります。KDC レルムには、Kerberos サーバの FQDN が記録されているサーバ（SRV）レコードを含むレルム名と同じ名前のゾーンが存在する必要があります。

SRV レコードで指定される Kerberos サーバ自体は、適切なゾーンの A レコードを持っている必要があります。また MTA 設定ファイルで指定されている Call Management Server（CMS; コール管理サーバ）も、適切なゾーンの A レコードを持っている必要があります。さらに、CMS は MTA の FQDN を解決することによってその MTA に到達するため、MTA 自体も適切なゾーンの A レコードを持っている必要があります。MTA の A レコードを作成する方法としては、ダイナミック DNS（DDNS）を使用することをお勧めします。DDNS の設定およびトラブルシューティングの詳細については、Cisco Network Registrar のマニュアルを参照してください。

KDC

KDC は、MTA の認証を行います。そのため、MTA の証明書を検査するとともに、KDC 自体の証明書を提示して MTA が KDC を認証できるようにする必要があります。また、DPE（プロビジョニング サーバ）と通信して、MTA がネットワークでプロビジョニングされていることを検証します。

PacketCable プロビジョニング サーバ

PacketCable プロビジョニング サーバは、MTA 設定ファイルの場所を MTA に伝達したり、SNMP 経由で MTA パラメータをプロビジョニングしたりします。MTA とプロビジョニング サーバの間のすべての通信で、SNMPv3 を使用します。SNMPv3 通信を開始するためのキーは、KDC との認証フェーズ中に MTA が取得します。プロビジョニング サーバの機能は、BAC アーキテクチャの DPE によって提供されます。

コール管理サーバ

コール管理サーバ（CMS）は、基本的にはソフト スイッチ、つまりコール エージェントです。追加の PacketCable 機能として、たとえばケーブル ネットワークの QoS を制御したりします。MTA は、PacketCable プロビジョニングに成功すると、Network Call Signaling（NCS; ネットワーク コール シグナリング）の Restart in Progress（RSIP; 再起動中）メッセージを CMS に送信します。

主要な変数

この項では、eMTA を適正にプロビジョニングするために必要とされる主な変数について説明します。

- 証明書（P.16-14）
- スコープ選択タグ（P.16-15）
- MTA 設定ファイル（P.16-15）

証明書

MTA_Root.cer ファイルには、MTA ルート証明書（正式な PacketCable MTA ルートをルートとする証明書）が含まれています。

プロビジョニングの対象となる MTA で必要とされるテレフォニー ルート証明書をあらかじめ把握しておく必要があります。実稼働ネットワークへの配備の際に、PacketCable の実稼働ルートをルートとするテレフォニー証明書を使用します。テスト環境で使用される PacketCable テスト ルートもあります。

KDC がそれ自体を MTA に対して認証するために使用する KDC 証明書のルートは、MTA に保存されているルート（PacketCable の実稼働ルートまたはテスト ルート）と同じテレフォニー ルートになっている必要があります。ほとんどの MTA ベンダーは Telnet または HTTP ログイン機能を備えたテスト イメージをサポートしているため、イネーブルになっているテレフォニー ルートを判別し、使用するルートを変更できます（ほとんどの場合、選択できるのは PacketCable の実在ルートとテスト ルートのどちらかのみです）。

最も一般的なシナリオでは、（*BPR_HOME/kdc/solaris/packetcable/certificates* ディレクトリから）次の証明書と一緒にロードした KDC を使用します。

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

最初の 4 つの証明書は、テレフォニー証明書チェーンを構成します。*MTA_Root.cer* ファイルには、MTA によって送信される証明書を検証するために、KDC が使用する MTA ルートが記述されています。

（注） KDC 証明書のインストールと管理の詳細については、P.14-3 の「PKCert.sh ツールの使用方法」を参照してください。

PacketCable テスト ルートを使用しているかどうかを判断するには、Windows で *CableLabs_Service_Provider_Root.cer* ファイルを開き、Subject OrgName エントリが **O = CableLabs** になっていることを確認するか、または Subject Alternative 名が **CN=CABLELABS GENERATED TEST ROOT FOR EQUIPMENT TEST PURPOSES ONLY** になっていることを確認します。

KDC 証明書（*KDC.cer*）には、使用するレルム名が記述されています。BAC（および対応する DNS ゾーン）で使用するように設定されているレルム名は、このレルム名と一致している必要があります。また、MTA 設定ファイルのレルム org 名は、テレフォニー ルートに含まれる組織名と一致している必要があります。

KDC 証明書には、対応する秘密鍵が記述されており、*BPR_HOME/kdc/solaris* ディレクトリにインストールする必要があります。通常、秘密鍵の名前は、KDC_private_key.pkcs8 または KDC_private_key_proprietary です。証明書を変更する場合は、秘密鍵も変更する必要があります。

スコープ選択タグ

ほとんどのシナリオにおいて、BAC は、スコープ選択タグの付いたスコープからのすべての DHCP 要求の処理に関係があります。スコープ選択タグは、BAC 管理者のユーザ インターフェイスの DHCP Criteria ページで指定される選択基準に一致します。スコープを BAC 処理に関連付けるためにクライアント クラスを使用することもできます。この関連付けは、必ずデバイスをプロビジョニングする前に行ってください。

MTA 設定ファイル

MTA 設定ファイルには、CMS の場所が記述されています。また、レルム名のエントリが必ず記述されています。この値は、使用中の証明書チェーンの値と一致する必要があります。

MTA 設定ファイル内の特定のテーブル エントリは、MTA に Option 122 で配信されたレルム名に基づいてインデックス付けされます。MTA 設定ファイル内のこのレルム名エントリは、Option 122 で配信されたレルム名と一致する必要があります。たとえば、Option 122 で配信されたレルム名が **DEF.COM** であった場合、MTA 設定ファイルの pktcMtaDevRealm テーブルのエントリは、68.69.70.46.67.79.77 などのようにこのレルム名の ASCII 符号化文字値（Cisco Broadband Configurator を使用する場合はドット区切りの 10 進形式）で構成されるサフィックスを使用してインデックス付けされます。Web 上には、この変換を容易に行うことができる無償の ASCII 変換ページが数多くあります。

トラブルシューティングのツール

PacketCable MTA デバイスのプロビジョニング仕様で規定されている 25 の eMTA セキュア プロビジョニングのステップを図 7-1 に示します。この項では、次のトピックについて取り上げます。

- ログ（P.16-15）
- Ethereal、SnifferPro、およびその他のパケット キャプチャ ツール（P.16-16）

ログ

情報を保持するために次のログ ファイルが使用されます。

- Network Registrar には、ログが 2 つ（*name_dhcp_1_log* および *name_dns_1_log*）あります。これらのログには、Network Registrar からの最新のロギング エントリが記録されます。DHCP または DNS に関連した問題の場合には、これらのファイルを調べてください。
- *BPR_HOME/kdc/logs/kdc.log* ファイルには、KDC と MTA のインタラクションすべてと、KDC と DPE のインタラクションが表示されます。

- *BPR_DATA/dpe/logs/dpe.log* ファイルには、SNMPv3 の MTA とのインタラクションに関連する主な手順が表示されます。

（注）コマンドライン インターフェイス（CLI）を使用して、SNMP、登録サーバ、および登録サーバの詳細メッセージのトレースを有効にすると、潜在的な PacketCable 問題のトラブルシューティングに役立ちます。適切なトラブルシューティング用のコマンドの使用方法の詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

Ethereal、SnifferPro、およびその他のパケット キャプチャ ツール

パケット キャプチャ ツールは、eMTA のトラブルシューティングに不可欠のツールです。CableLabs がパッケージ化している Ethereal バージョンには、PacketCable に固有のパケット デコーダが数多く含まれています。たとえば、Kerberos の AS パケットや AP パケットなどがあります。

- 障害の原因が DHCP に関連していると疑われる場合は、CMTS ケーブル インターフェイスの IP アドレスと DHCP サーバの IP アドレスを送信元または宛先とするパケットをフィルタリングしながらパケットをキャプチャします。
- 障害の原因が DHCP 以降の 25 のステップのいずれかに関連していると疑われる場合は、eMTA の IP アドレスを送信元または宛先とするパケットをすべてフィルタリングします。この方法により、図 7-1 に示されるプロビジョニングのステップ 5 ～ 25 を非常に簡単にトレースできます。

トラブルシューティングのシナリオ

表 16-3 に示すシナリオは、eMTA が関係する可能性のある障害です。

表 16-3　トラブルシューティングのシナリオ

予想される問題	考えられる原因	対処方法
KDC が起動しない。	KDC 証明書が秘密鍵に対応していません。	証明書と秘密鍵を確実に一致させます。
	KDC ライセンスの期限が切れているか、または消失しています。	KDC ライセンスを *BPR_HOME/kdc* ディレクトリに復元します。
MTA デバイスが BAC Devices ページに表示されない。	不正なケーブル ヘルパー アドレスが設定されている可能性があります。	ヘルパー アドレスを修正します。
	スコープ選択タグが BAC ユーザ インターフェイスで選択された DHCP 基準と一致しません。	関係する MTA について、MTA スコープ選択タグが、作成された PacketCable DHCP 基準のスコープ選択タグと一致することを BAC で確認します。
	Network Registrar 拡張ポイントが正しくインストールされていません。	Network Registrar 拡張ポイントを再インストールします。『*Installation and Setup Guide for Cisco Broadband Access Center 4.0*』を参照してください。
	ケーブル モデム部分が Option 122 を受信しませんでした。	ケーブル モデム部分のスコープのタグが、BAC に対して設定されている DOCSIS DHCP 基準と一致することを確認します。

表 16-3　トラブルシューティングのシナリオ（続き）

予想される問題	考えられる原因	対処方法
MTA デバイスが DHCP オファーを受け入れず、DHCP フローのサイクルを続ける。	無効な DHCP オプションが設定されています。	スコープ ポリシーに DNS サーバ オプションが含まれていることを確認するか、または *cnr_ep.properties* ファイルにプライマリ DNS サーバとセカンダリ DNS サーバのエントリが含まれていることを確認します。
	DHCP オファーが、ケーブル モデム部分の Option 122 のサブオプション 1 で指定されている DHCP サーバとは異なるサーバから送信された可能性があります。	*cnr_ep.properties* ファイルを調べ、メインとバックアップの DHCP サーバが正しく設定されていることを確認します。
kdc.log ファイルと Ethereal トレースの両方で、MTA デバイスが KDC に問い合せていないことが示される。	*cnr_ep.properties* ファイルと MTA スコープ ポリシーの一方または両方で、不正な DNS サーバが指定されています。	*cnr_ep.properties* の DNS サーバを確認または訂正します。
	Kerberos レルムのゾーンが存在しないか、または正しく設定されていません。	レルムと同じ名前のゾーンが作成されており、「_kerberos._udp 0 0 88 *KDC FQDN*」形式の「SRV」レコードが含まれていることを確認します。
	KDC の「A」レコード エントリが存在しないか、または正しくありません。	Kerberos ゾーンの「SRV」レコードに含まれている FQDN の「A」レコードが存在することを確認します。
	DPE FQDN を解決できません。	*dpe.properties* の provFQDNs エントリに、DPE の正しい FQDN と IP があることを確認します。

表 16-3　トラブルシューティングのシナリオ（続き）

予想される問題	考えられる原因	対処方法
Kerberos の AS 要求中に、KDC が障害を報告する。	MTA 証明書が KDC で使用される MTA ルートと一致しません。	*MTA_Root.cer* を稼働システムで使用されている証明書と比較することにより、*MTA_Root.cer* が正しいことを確認します。 正しい場合、MTA 自体で証明書の問題が発生している可能性があります。このような状況は非常にまれですが、そうなった場合には、MTA の製造業者に連絡してください。
	KDC によるプロビジョニング サーバへの FQDN ルックアップに失敗しました。そのデバイスは、BAC でまだプロビジョニングされていない可能性があります。	デバイスが表示されることを確認します。サービス クラスと DHCP 基準の両方が指定されている必要があります。
	クロック スキュー エラーです。詳細については、P.3-7 の「PacketCable ワークフロー」を参照してください。	すべての BAC ネットワーク要素が、NTP を介してクロック同期されていることを確認します。『*Broadband Access Center DPE CLI Reference 4.0*』を参照してください。
	KDC と DPE の間に不一致が存在する可能性があります。 **（注）** 他のデバイスが正しくプロビジョニングされている場合は、これが問題の原因とは考えられません。	*BPR_HOME/kdc/solaris/keys* ディレクトリに次の 3 つのエントリがあることを確認します。 • mtafqdnmap,dpe.abc.com@DEF.COM • mtaprovsrvr,dpe.abc.com@DEF.COM • krbtgt,DEF.COM@DEF.COM ご使用のシステムの DPE FQDN とレルム名は、この例の場合とは異なります。これらのエントリの内容は、dpe.properties の「KDCServiceKey」エントリまたは KeyGen ユーティリティを使用して生成されたキーのいずれかのエントリと一致している必要があります。
KDC により、AS 要求 / 応答（図 7-1 のステップ 9 と 10）で成功と報告されるが、TMA デバイスがステップ 9 より先に進まない。	MTA でロードまたはイネーブルにされているテレフォニー ルートと、KDC にロードされているテレフォニー ルートの間に証明書の不一致があります。	MTA と KDC の証明書を確認してください。
	非常にまれなことですが、テレフォニー証明書チェーンが破損している可能性があります。 **（注）** 他のデバイスが正しくプロビジョニングされている場合、これは問題の原因ではありません。	MTA で正しい証明書がロードまたはイネーブルされていることを確認します。正しくプロビジョニングできるデバイスがない場合は、KDC にある別の証明書を試してください。

表 16-3　トラブルシューティングのシナリオ（続き）

予想される問題	考えられる原因	対処方法
AP 要求 / 応答（図 7-1 のステップ 14）で障害が発生する。	クロック スキュー エラーです。詳細については、P.3-7 の「PacketCable ワークフロー」を参照してください。	すべての BAC ネットワーク要素が、NTP を介してクロック同期されていることを確認します。『*Broadband Access Center DPE CLI Reference*』を参照してください。
	プロビジョニング サーバの FQDN を解決できません。	プロビジョニング サーバ（DPE）の DNS エントリが正しいことを確認します。 dpe.properties provFQDNs エントリに、プロビジョニング サーバ（DPE）の正しい FQDN と IP があることを確認します。
	MTA から DPE へのルートがありません。	ルーティング問題を修正します。
MTA デバイスが設定ファイルの TFTP 要求を発行しない。	DPE で実行されている TFTP サーバへのルートがありません。	ルーティング問題を修正します。
MTA デバイスが TFTP 設定ファイルを受信しない。	DPE で設定ファイルがキャッシュされません。	次のプロビジョニングが試行されてファイルがキャッシュされるまで待ちます。これでキャッシュされない場合は、MTA をリセットします。
	Network Registrar の MTA スコープ ポリシーに、矛盾する TFTP サーバ オプションが含まれています。	BAC が TFTP サーバの DPE アドレスを挿入するため、ポリシーからこのオプションを安全に削除できます。
MTA デバイスは設定ファイルを受信するが、DPE は *dpe.log* ファイルにある SNMP Inform（図 7-1 のステップ 25）の受信に失敗する。	次のいずれかの状況が考えられます。 • 設定ファイルの内部での矛盾。 • テレフォニー証明書チェーンのレルム起点との矛盾。 • Option 122 でのレルム名との矛盾。	MTA 設定ファイルに整合性があることを確認します。
RSIP が送信されなくても、MTA デバイスが成功と報告する（図 7-1 のステップ 25）。	MTA が、MTA 設定ファイルで指定されている CMS FQDN の IP アドレスを解決できません。	CMS の DNS エントリが存在することを確認します。
	MTA が CMS の IP アドレスに到達できません。これは、ルートが設定されていないことを示します。	すべてのルーティング問題を解決します。
MTA デバイスが、CMS サービスを受けるために KDC に再び問い合せるにもかかわらず、成功と報告する（図 7-1 のステップ 25）。	MTA 設定ファイルが誤ったケーブル モデムを示しています。	設定ファイルを訂正するか、設定ファイルのリストにある FQDN を使用するように Cisco BTS 10200 を再設定します。
	MTA 設定ファイルの pktcMtaDevCmsIPsecCtrl 値が存在しないか、1 に設定されています。これは、MTA がセキュア NCS コール シグナリングを実行すること、または CMS FQDN の ASCII サフィックスと一致しない ASCII サフィックスを使用することを意味します。	設定ファイルを訂正します。セキュア シグナリングを実行する場合は、サポートのために必要な手順を実行して KDC と BTS を設定します。

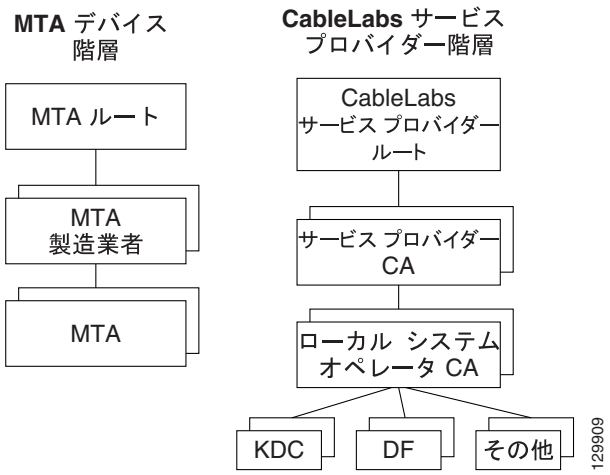
表 16-3　トラブルシューティングのシナリオ（続き）

予想される問題	考えられる原因	対処方法
MTA デバイスが成功と報告し（図 7-1 のステップ 25）、RSIP を送信するが、ソフト スイッチからの応答がないか、応答でエラーが返される。	MTA が Cisco BTS 10200 上でプロビジョニングされていないか、または正しくプロビジョニングされていません。	Cisco BTS 10200 で MTA をプロビジョニングします。
	eMTA DNS エントリが存在しません。	eMTA の正しい DNS ゾーンにエントリを配置します。ダイナミック DNS の使用をお勧めします。DDNS のイネーブル化の詳細については、Cisco Network Registrar のマニュアルを参照してください。

証明書信頼階層

BAC PacketCable に関係する証明書階層には、図 16-1 に示すように、MTA デバイス証明書階層と CableLabs サービス プロバイダー証明書階層の 2 つがあります。

図 16-1　PacketCable 証明書階層



PacketCable を BAC に実装する前に、次の技術ドキュメントの内容に精通しておいてください。

- *RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile*
- *DOCSIS Baseline Privacy Plus Interface Specification（SP-BPI+-I11-040407、2004 年 4 月 7 日）*

（注）Euro PacketCable では PacketCable のセキュリティ仕様 [PKT-SP-SEC-I08-030415] を使用していますが、Euro-PacketCable 環境で使用されるデジタル証明書に関連して、いくつかの変更を行う必要があります。Euro PacketCable と PacketCable をできるだけ類似した状態に保つため、Euro PacketCable ではすべての PacketCable セキュリティ技術を使用しており、その中にはセキュリティ仕様 [PKTSP-SEC-I08-030415] の新しいリビジョンも含まれます。

PacketCable 証明書とは異なる Euro-PacketCable 証明書の要素を以下の表に示します。

Euro PacketCable では、Euro-PacketCable 証明書が唯一有効な証明書です。PacketCable 証明書を参照する PacketCable の [PKT-SP-SEC-I08-030415] に記載されているすべての要件は、Euro-PacketCable 証明書の対応する要件に変更されます。

Euro-PacketCable 準拠 eMTA では、ケーブル モデムの非揮発性メモリの中に、DOCSIS CVC CA の公開鍵の代わりに、Euro-DOCSIS ルート CVC CA の公開鍵が保存されている必要があります。Euro PacketCable 準拠の独立型 MTA では、tComLabs CVC ルート証明書と tComLabs CVC CA 証明書が非揮発性メモリに保存されている必要があります。製造業者の CVC は、証明書チェーンを検査することで検証されます。

証明書の検証

PacketCable 証明書の検証には、一般に、証明書チェーン全体の検証が含まれます。たとえば、プロビジョニング サーバが MTA デバイス証明書を検証する場合、次の証明書チェーンが検証されます。

MTA ルート証明書 + MTA 製造業者証明書 + MTA デバイス証明書

MTA 製造業者証明書の署名は MTA ルート証明書によって検証され、MTA デバイス証明書の署名は MTA 製造業者証明書の署名によって検証されます。MTA ルート証明書は自己署名され、プロビジョニング サーバに前もって知らされます。MTA ルート証明書内の公開鍵は、この同じ証明書の署名を検証するために使用されます。

通常、チェーンの最初の証明書は、通信経路を通して送信される証明書チェーンに明示的に指定されていません。最初の証明書が明示的に含まれている場合は、検証する側にあらかじめ知らされている必要があり、証明書のシリアル番号、有効期間、および署名の値などの例外を除いて、証明書に変更が一切ないようにする必要があります。既知の CableLabs サービス プロバイダーのルート証明書と比較して、通信経路を通して渡された CableLabs サービス プロバイダーのルート証明書に変更があると、比較を行うデバイスは証明書の検証に必ず失敗します。

証明書チェーン検証の実際のルールは、RFC 2459 に完全に準拠している必要があります。RFC 2459 では、証明書チェーン検証を証明書パス検証と呼んでいます。一般に、X.509 証明書は、証明書の発行者名がもう一方の証明書のサブジェクト名と一致しているかどうかを判定するための自由なルール セットをサポートしています。このルール セットでは、2 つの名前フィールドのバイナリ比較が一致していることを示さなくても、それらの名前フィールドが一致すると宣言される場合があります。RFC 2459 では、実装環境において、単純なバイナリ比較を使用して一致または不一致を宣言することができるように、認証局で名前フィールドの符号化を制限するよう推奨しています。

PacketCable のセキュリティは、この推奨事項に従っています。したがって、PacketCable 証明書の DER 符号化された tbsCertificate.issuer フィールドが、その発行者の証明書の DER 符号化された tbsCertificate.subject フィールドと完全に一致している必要があります。実装環境では、DER 符号化された tbsCertificate.issuer フィールドど tbsCertificate.subject フィールドのバイナリ比較を実行することによって、発行者名とサブジェクト名を比較することができます。

次の項では、必要な証明書チェーンを指定します。それらの証明書チェーンを使用して、図 16-1 に示す PacketCable 証明書信頼階層の（最下位の）リーフ ノードに存在する各証明書を検証する必要があります。

入れ子になっている有効期間は検査されず、故意に実行されてはいません。このため、証明書の有効期間は、それを発行した証明書の有効期間内に入る必要はありません。

MTA デバイス証明書階層

デバイス証明書階層は、DOCSIS1.1/BPI+ 階層のデバイス証明書階層をそのままミラーリングしています。ルートは CableLabs 発行の PacketCable MTA ルート証明書で、このルート証明書は、一連の製造業者証明書の発行元証明書として使用されます。製造業者証明書は、個々のデバイス証明書に署名するために使用されます。

以降の表に示す情報には、RFC 2459 に従った必須フィールドの PacketCable 固有の値が含まれています。これらの PacketCable 固有の値は、表 16-4 の情報に従って指定する必要があります。ただし、有効期間の値は、それぞれの表で指定されている値にします。PacketCable での必須フィールドが明示的に示されていない場合は、RFC 2459 のガイドラインに従ってください。

MTA ルート証明書

この証明書は、MTA ルート証明書、MTA 製造業者証明書、および MTA デバイス証明書で構成される証明書チェーンの一部として検証する必要があります。

表 16-4 に、MTA ルート証明書に関係する値のリストを示します。

表 16-4　MTA ルート証明書

MTA ルート証明書		
Subject Name Form	**PacketCable** C=US O=CableLabs OU=PacketCable CN=PacketCable Root Device Certificate Authority	**Euro PacketCable** C=BE O=tComLabs OU=Euro-PacketCable CN=Euro-PacketCable Root Device Certificate Authority
Intended Usage	この証明書は、MTA 製造業者証明書に署名するために使用されるとともに、KDC によって使用されます。この証明書は MTA によって使用されることがないため、MTA MIB には表示されません。	
Signed By	自己署名	
Validity Period	20 年以上。この証明書を再発行する必要が生じることがないように、十分な長さの有効期間が設定されています。	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true, pathLenConstraint=1)	

MTA 製造業者証明書

この証明書は、MTA ルート証明書、MTA 製造業者証明書、および MTA デバイス証明書で構成される証明書チェーンの一部として検証する必要があります。州、市、および製造業者の施設は、オプションの属性です。製造業者は、複数の製造業者証明書を備えることがあり、製造業者ごとに 1 つ以上の証明書が存在する場合もあります。同じ製造業者の証明書すべてを、製造時または現地でのアップデート中に各 MTA に提供することができます。MTA は、MTA デバイス証明書にある発行者名を MTA 製造業者証明書にあるサブジェクト名と照合して、使用する適切な証明書を選択する必要があります。存在する場合は、RFC 2459 で規定されているように、デバイス証明書の authorityKeyIdentifier が製造業者証明書の subjectKeyIdentifier と一致する必要があります。O および CN の *CompanyName* フィールドは、その 2 つのインスタンス間で異なる場合があります。

表 16-5 に、MTA 製造業者証明書に関係する値のリストを示します。

表 16-5　MTA 製造業者証明書

MTA 製造業者証明書		
Subject Name Form	**PacketCable** C=US O=CableLabs OU=PacketCable CN=PacketCable Root Device Certificate Authority	**Euro PacketCable** C=*Country of Manufacturer* O=*Company Name* [stateOrProvinceName = *State/Province*] [localityName=*City*] OU=Euro-PacketCable [organizationalUnitName= *Manufacturing Location*] CN=*Company Name* Euro-PacketCable CA
Intended Usage	この証明書は、各 MTA 製造業者に対して発行され、PacketCable セキュリティ仕様の規定どおり（製造時または現地でのアップデート中に）セキュア コード ダウンロードの一環として各 MTA にインストールできます。この証明書は、MTA MIB 中に読み取り専用パラメータとして表示されます。この証明書は、KDC による認証中に、MTA デバイスのアイデンティティ（MAC アドレス）を認証するために、MTA デバイス証明書と一緒に使用されます。	
Signed By	MTA ルート証明書の CA	
Validity Period	20 年	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier[n,m], authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*), basicConstraints[c,m](cA=true, pathLenConstraint=0)	

MTA デバイス証明書

この証明書は、MTA ルート証明書、MTA 製造業者証明書、および MTA デバイス証明書で構成される証明書チェーンの一部として検証する必要があります。州、市、および製造業者の施設は、オプションの属性です。MAC アドレスは、6 組のコロン区切り 16 進数（「00:60:21:A5:0A:23」など）として指定する必要があります。16 進数のアルファベット文字（A ～ F）は、大文字で表記する必要があります。MTA デバイス証明書は、置換または更新しないでください。

表 16-6 に、MTA デバイス証明書に関係する値のリストを示します。

表 16-6　MTA デバイス証明書

MTA デバイス証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*], OU=PacketCable [OU=*Product Name*] [OU=*Manufacturer's Facility*] CN=*MAC Address*	**Euro PacketCable** C=*Country of Manufacturer* O=*Company Name* [ST=*State/Province*] [L=*City*] OU=Euro-PacketCable [OU=*Product Name*] [OU=*Manufacturing Location*] CN=*MAC Address*
Intended Usage	この証明書は、MTA 製造業者によって発行され、製造時にインストールされます。プロビジョニング サーバは、この証明書をアップデートできません。この証明書は、MTA MIB 中に読み取り専用パラメータとして表示されます。この証明書は、プロビジョニング中に MTA デバイスのアイデンティティ（MAC アドレス）を認証するために使用されます。	
Signed By	MTA 製造業者証明書の CA	
Validity Period	20 年以上	
Modulus Length	1024、1536、または 2048	
Extensions	keyUsage[c,o](digitalSignature, keyEncipherment) authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*)	

MTA 製造業者コード検証証明書

eMTA のコード検証証明書（CVC）仕様は、DOCSIS 仕様 SP-BPI+-I11-040407 で指定されている DOCSIS 1.1 CVC と同一の仕様にする必要があります。

CableLabs サービス プロバイダー証明書階層

サービス プロバイダー証明書階層のルートは、CableLabs 発行の CableLabs サービス プロバイダー ルート証明書です。この証明書は、一連のサービス プロバイダー証明書の発行元証明書として使用されます。サービス プロバイダーの証明書は、オプションのローカル システム証明書に署名するために使用されます。ローカル システム証明書が存在する場合は、補助装置証明書に署名するためにその証明書が使用されます。存在しない場合には、サービス プロバイダーの CA が補助証明書に署名します。

表 16-7 の情報には、RFC 2459 での必須フィールドに対する固有の値が含まれています。それらの固有値を使用する必要があります。必須フィールドがリストに含まれていない場合は、RFC 2459 のガイドラインに厳密に従う必要があります。

CableLabs サービス プロバイダー ルート証明書

Kerberos キー管理を実行できるようにするには、Kerberos プロトコルに対する PKINIT 拡張を使用して、事前に MTA と KDC で相互認証を実行する必要があります。MTA は、KDC 証明書チェーンを含んだ PKINIT Reply メッセージを受信した後に KDC を認証します。KDC の認証を行う場合、MTA は、CableLabs サービス プロバイダー ルート CA が署名した KDC のサービス プロバイダー証明書を含む KDC 証明書チェーンを検証します。

表 16-7 に、CableLabs サービス プロバイダー ルート 証明書に関係する値のリストを示します。

表 16-7　CableLabs サービス プロバイダー ルート証明書

CableLabs サービス プロバイダー ルート証明書		
Subject Name Form	**PacketCable** C=US O=CableLabs CN=CableLabs Service Provider Root CA	**Euro PacketCable** C=BE O=tComLabs CN=tComLabs Service Provider Root CA
Intended Usage	この証明書は、サービス プロバイダー CA 証明書に署名するために使用されます。この証明書は、製造時に各 MTA にインストールされるか、または PacketCable セキュリティ仕様の規定どおりセキュア コード ダウンロードによってインストールされ、プロビジョニング サーバがアップデートすることはできません。このルート証明書および対応する公開鍵は、いずれも MTA MIB に表示されることはありません。	
Signed By	自己署名	
Validity Period	20 年以上。この証明書を再発行する必要が生じることがないように、十分な長さの有効期間が設定されています。	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true)	

サービス プロバイダー CA 証明書

これはサービス プロバイダーが保持する証明書で、CableLabs サービス プロバイダー ルート CA によって署名されます。CableLabs サービス プロバイダー ルート証明書、テレフォニー サービス プロバイダー証明書、オプションのローカル システム証明書、およびエンドエンティティ サーバ証明書が含まれる証明書チェーンの一部として検証されます。認証する側のエンティティは、通常はすでに CableLabs サービス プロバイダー ルート証明書を所有しており、この証明書は、証明書チェーンの残りの部分とともに転送されることはありません。

サービス プロバイダー CA 証明書が常に明示的に証明書チェーンに含まれているため、サービス プロバイダーは、自身の証明書を柔軟に変更でき、この証明書チェーンを検証する各エンティティ（たとえば、MTA は PKINIT Reply を検証します）を再設定する必要はありません。サービス プロバイダー CA 証明書を変更するたびに、CableLabs サービス プロバイダー ルート証明書を使用してその署名を検証する必要があります。ただし、同じサービス プロバイダーの新しい証明書では、SubjectName の OrganizationName 属性を以前と同じ値に保つ必要があります。O および CN にある *Company* フィールドは、その 2 つのインスタンス間で異なる場合があります。

表 16-8 に、CableLabs サービス プロバイダー CA 証明書に関係する値のリストを示します。

表 16-8　CableLabs サービス プロバイダー CA 証明書

CableLabs サービス プロバイダー ルート証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company* CN=*Company* CableLabs Service Provider CA	**Euro PacketCable** C=*Country* O=*Company* CN=*Company* tComLabs Service Provider CA
Intended Usage	この証明書は、サービス プロバイダー CA 証明書に署名するために使用されます。この証明書は、製造時に各 MTA にインストールされるか、または PacketCable セキュリティ仕様の規定どおりセキュア コード ダウンロードによってインストールされ、プロビジョニング サーバがアップデートすることはできません。このルート証明書および対応する公開鍵は、いずれも MTA MIB に表示されることはありません。	
Signed By	自己署名	
Validity Period	20 年以上。この証明書を再発行する必要が生じることがないように、十分な長さの有効期間が設定されています。	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign cRLSign), subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true)	

ローカル システム CA 証明書

サービス プロバイダー CA は、ローカル システム CA と呼ばれる地域別の認証局（対応するローカル システム証明書を発行する）に証明書の発行を委任することがあります。ネットワーク サーバは、同じサービス プロバイダーの地域別の認証局間を自由に移動できます。したがって、MTA MIB にはローカル システム証明書に関する情報は含まれていません（ローカル システム証明書により、MTA が特定地域内の KDC に制限される可能性があります）。

表 16-9 に、ローカル システム CA 証明書に関係する値のリストを示します。

表 16-9　ローカル システム CA 証明書

ローカル システム CA 証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company* OU=*Local System Name* CN=*Company* CableLabs Local System CA	**Euro PacketCable** C=*Country* O=*Company* OU=*Local System Name* CN=*Company* tComLabs Local System CA
Intended Usage	サービス プロバイダー CA は、ローカル システム CA と呼ばれる地域別の認証局（対応するローカル システム証明書を発行する）に証明書の発行を委任することがあります。ネットワーク サーバは、同じサービス プロバイダーの地域別の認証局間を自由に移動できます。したがって、MTA MIB にはローカル システム証明書に関する情報は含まれていません（ローカル システム証明書により、MTA が特定地域内の KDC に制限される可能性があります）。	

表 16-9　ローカル システム CA 証明書 （続き）

ローカル システム CA 証明書	
Signed By	サービス プロバイダー CA 証明書
Validity Period	20 年
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier[n,m], authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*), basicConstraints[c,m](cA=true, pathLenConstraint=0)

運用上の補助証明書

この項に示すすべての証明書は、ローカル システム CA またはサービス プロバイダー CA によって署名されます。この標準には、将来的に他の補助証明書が追加されることがあります。

KDC 証明書

この証明書は、CableLabs サービス プロバイダー ルート証明書、サービス プロバイダー CA 証明書、および補助デバイス証明書で構成される証明書チェーンの一部として検証する必要があります。PKINIT 仕様では、KDC 証明書に subjectAltName v.3 証明書拡張を含めるよう規定しています。この証明書拡張の値は、KDC の Kerberos プリンシパル名にする必要があります。

表 16-10 に、KDC 証明書に関係する値のリストを示します。

表 16-10　KDC 証明書

鍵発行局証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company*, [OU=*Local System Name*] OU= CableLabs Key Distribution Center CN=*DNS Name*	**Euro PacketCable** C=*Country* O=*Company* [OU=*Local System Name*] OU=tComLabs Key Distribution Center CN=*DNS Name*
Intended Usage	KDC サーバのアイデンティティを PKINIT 交換中に MTA に対して認証すること。この証明書は PKINIT 応答の中で MTA に渡されるため、MTA MIB には含まれておらず、プロビジョニング サーバがアップデートおよび照会することはできません。	
Signed By	サービス プロバイダー CA 証明書またはローカル システム証明書	
Validity Period	20 年	
Modulus Length	1024、1536、または 2048	
Extensions	keyUsage[c,o](digitalSignature)authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*)subjectAltName[n,m]	

配信機能（DF）

この証明書は、CableLabs サービス プロバイダー ルート証明書、サービス プロバイダー CA 証明書、および補助デバイス証明書で構成される証明書チェーンの一部として検証する必要があります。この証明書は、（電子サーベイランスで使用される）DF 間でのフェーズ 1 IKE ドメイン間交換に署名するために使用されます。Local System Name はオプションですが、ローカル システム CA がこの証明書に署名する場合は必須です。IP アドレスは、245.120.75.22 などの標準的なドット付き 4 数字列表記で指定する必要があります。

表 16-11 に、DF 証明書に関係する値のリストを示します。

表 16-11　DF 証明書

DF 証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company* [OU=*Local System Name*] OU=PacketCable Electronic Surveillance CN=*IP address*	**Euro PacketCable** C=*Country* O=*Company* [OU=*Local System Name*] OU=Euro-PacketCable Electronic Surveillance CNe=*IP address*
Intended Usage	IKE キー管理を認証するために、1 組の DF 間で IPsec セキュリティ アソシエーションを確立するのに使用されます。これらのセキュリティ アソシエーションは、合法的に傍聴されているサブジェクトが、新しい傍聴サーバ（DF）に転送される必要のあるコール情報を含んだコール メッセージとイベント メッセージを転送するときに使用されます。	
Signed By	サービス プロバイダー CA 証明書またはローカル システム CA 証明書	
Validity Period	20 年	
Modulus Length	2048	
Extensions	keyUsage[c,o](digitalSignature) authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*) subjectAltName[n,m] (dNSName=*DNSName*)	

PacketCable サーバ証明書

これらの証明書は、CableLabs サービス プロバイダー ルート証明書、サービス プロバイダー証明書、ローカル システム オペレータ証明書（使用されている場合）、および補助デバイス証明書で構成される証明書チェーンの一部として検証する必要があります。これらの証明書は、PacketCable システムの各種サーバを識別するために使用されます。たとえば、フェーズ 1 IKE 交換に署名するため、または PKINIT 交換を認証するために使用されることがあります。Local System Name はオプションですが、ローカル システム CA がこの証明書に署名する場合は必須です。IP アドレスの値は、245.120.75.22 などの標準的なドット区切り 10 進表記で指定する必要があります。DNS Name の値は、device.packetcable.com などの完全修飾ドメイン名（FQDN）で指定する必要があります。

表 16-12 に、PacketCable Server 証明書に関係する値のリストを示します。

表 16-12　PacketCable サーバ証明書

PacketCable サーバ証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company* OU=PacketCable OU=[*Local System Name*] OU=*Sub-System Name* CN=*Server Identifier*[:*Element ID*] *Server Identifier* の値は、サーバの FQDN または IP アドレスにする必要があり、オプションで、その値の後にコロン（:）（前後にスペースなし）と Element ID を続けることができます。 *Element ID* は、課金イベント メッセージに表示される ID です。イベント メッセージを生成できるすべてのサーバの証明書に含まれている必要があります。このようなサーバには、CMS、CMTS、および MGC があります。[8] は、5 オクテット右揃えの、空白文字が入力された、ASCII 符号化数値文字列として Element ID を定義します。証明書で使用するために Element ID を変換するときには、空白文字を ASCII の 0（0x48）に変換する必要があります。 たとえば、CMTS の Element ID が 311 で、IP アドレスが 123.210.234.12 の場合、CMTS の通常名は「123.210.234.12:00311」となります。 *Sub-System Name* の値は、次のいずれかにする必要があります。 • ボーダー プロキシの場合：bp • ケーブル モデム ターミネーション システムの場合：cmts • コール管理サーバの場合：cms • メディア ゲートウェイの場合：mg • メディア ゲートウェイ コントローラの場合：mgc • メディア プレーヤーの場合：mp • メディア プレーヤー コントローラの場合：mpc • プロビジョニング サーバの場合：ps • レコード記録サーバの場合：rks • シグナリング ゲートウェイの場合：sg	**Euro PacketCable** C=*Country* O=*Company* OU=Euro-PacketCable [OU=*Local System Name*] OU=*Sub-system Name* CN=*Server Identifier*[:*Element ID*] commonName の追加仕様については、[PKT-SP-SEC-IO8-030415] を参照してください。
Intended Usage	これらの証明書は、PacketCable システムの各種サーバを識別するために使用されます。たとえば、フェーズ 1 IKE 交換に署名するため、または PKINIT 交換でデバイスを認証するために使用される場合があります。	
Signed By	テレフォニー サービス プロバイダー証明書またはローカル システム証明書	

表 16-12　PacketCable サーバ証明書 （続き）

PacketCable サーバ証明書	
Validity Period	MSO ポリシーにより設定
Modulus Length	2048
Extensions	keyUsage[c,o](digitalSignaturekeyEncipherment) authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA cert*) subjectAltName[n,m](dNSName=*DNSName* | iPAddress=*IP AddressName*) KeyUsage タグはオプションです。使用する場合は、このタグをクリティカルとしてマークする必要があります。特に説明のない限り、次の subjectAltName 拡張には、サブジェクトの CN フィールドで指定された対応する名前値が含まれている必要があります。

CMS 証明書の CN 属性値は、Element ID にする必要があります。subjectAltName 拡張には、CMS の IP アドレスまたは FQDN のいずれかが含まれている必要があります。CMTS 証明書の CN 属性値は、Element ID にする必要があります。subjectAltName 拡張には、CMTS の IP アドレスまたは FQDN のいずれかが含まれている必要があります。

MGC 証明書の CN 属性値は、Element ID にする必要があります。subjectAltName 拡張には、MGC の IP アドレスまたは FQDN のいずれかが含まれている必要があります。

証明書失効

現時点では、PacketCable の仕様範囲外です。

コード検証証明書階層

CableLabs コード検証証明書（CVC）PKI は汎用性を備えており、CVC を必要とするすべての CableLabs プロジェクトに適用できます。つまり、基本インフラストラクチャをあらゆる CableLabs プロジェクトで再利用することができます。必要となるエンドエンティティ証明書はプロジェクトによって異なる場合がありますが、エンドエンティティ証明書が重複している場合は、1 つのエンドエンティティ証明書を使用してその重複をサポートできます。

CableLabs CVC 階層は、eMTA には適用されません。

CVC の共通要件

すべてのコード検証証明書に対して、次の要件が適用されます。

- 証明書は、DER 符号化されている必要がある。
- 証明書は、バージョン 3 にする必要がある。
- 証明書は、以降の各表で指定されている拡張を含んでいる必要があり、その他の拡張を含んでいてはならない。
- 公開指数は、F4（10 進数の 65537）である必要がある。

CableLabs コード検証ルート CA 証明書

この証明書は、CableLabs コード検証ルート CA 証明書、CableLabs コード検証 CA 証明書、およびコード検証証明書で構成される証明書チェーンの一部として検証する必要があります。証明書の検証方法の詳細については、P.16-21 の「証明書の検証」を参照してください。

表 16-13 に、CableLabs コード検証ルート CA 証明書に関係する値のリストを示します。

表 16-13　CableLabs コード検証ルート CA 証明書

CableLabs コード検証ルート CA 証明書		
Subject Name Form	**PacketCable** C=US O=CableLabs CN=CableLabs CVC Root CA	**Euro PacketCable** C = BE O = tComLabs CN = tComLabs CVC Root CA
Intended Usage	この証明書は、コード検証 CA 証明書に署名するために使用されます。この証明書は、製造時に S-MTA の非揮発性メモリに保存される必要があります。	
Signed By	自己署名	
Validity Period	20 年以上	
Modulus Length	2048	
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectkeyidentifier [n,m] basicConstraints[c,m](cA=true)	

CableLabs コード検証 CA 証明書

CableLabs コード検証 CA 証明書は、CableLabs コード検証ルート CA 証明書、CableLabs コード検証 CA 証明書、およびコード検証証明書で構成される証明書チェーンの一部として検証する必要があります。証明書の検証方法の詳細については、P.16-21 の「証明書の検証」を参照してください。CableLabs コード検証 CA は、複数存在する場合があります。S-MTA は、同時に 1 つの CableLabs CVC CA をサポートする必要があります。

表 16-14 に、CableLabs コード検証 CA 証明書に関係する値のリストを示します。

表 16-14　CableLabs コード検証 CA 証明書

CableLabs コード検証 CA 証明書		
Subject Name Form	**PacketCable** C=US O=CableLabs CN=CableLabs CVC CA	**Euro PacketCable** C = BE O = tComLabs CN = tComLabs CVC CA
Intended Usage	この証明書は、CableLabs コード検証ルート CA によって CableLabs に発行されます。この証明書がコード検証証明書を発行します。この証明書は、製造時に S-MTA の非揮発性メモリに保存される必要があります。	
Signed By	CableLabs コード検証ルート CA	
Validity Period	CableLabs ポリシーにより設定	

表 16-14　CableLabs コード検証 CA 証明書 （続き）

CableLabs コード検証 CA 証明書	
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectKeyIdentifier[n,m] authorityKeyIdentifier [n,m] basicConstraints [c,m](cA=true, pathLenConstraint=0)

製造業者コード検証証明書

CableLabs コード検証 CA は、認可された各製造業者に対してこの証明書を発行します。この証明書は、セキュアなソフトウェア ダウンロードのために CATV 事業者により設定されたポリシーで使用されます。

表 16-15 に、製造業者コード検証証明書に関係する値のリストを示します。

表 16-15　製造業者コード検証証明書

製造業者コード検証証明書		
Subject Name Form	**PacketCable** C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*] CN=*Company Name* Mfg CVC	**Euro PacketCable** C=*Country* O=*Company Name* [ST=*state/province*] [L=*City*] CN=*Company Name* Mfg CVC
Intended Usage	CableLabs コード検証 CA は、認可された各製造業者に対してこの証明書を発行します。この証明書は、セキュアなソフトウェア ダウンロードのために CATV 事業者により設定されたポリシーで使用されます。	
Signed By	CableLabs コード検証 CA	tComLabs コード検証 CA 証明書
Validity Period	CableLabs ポリシーにより設定	
Modulus Length	1024, 1536, 2048	
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m]	

Organization の Company Name は、Common Name の Company Name と異なる場合があります。

サービス プロバイダー コード検証証明書

サービス プロバイダー コード検証証明書は、CableLabs コード検証ルート CA 証明書、CableLabs コード検証 CA 証明書、およびサービス プロバイダー コード検証証明書で構成される証明書チェーンの一部として検証する必要があります。証明書の検証方法の詳細については、P.16-21 の「証明書の検証」を参照してください。

表 16-16 に、サービス プロバイダー コード検証証明書に関係する値のリストを示します。

表 16-16　サービス プロバイダー コード検証証明書

サービス プロバイダー コード検証証明書		
Subject Name Form	C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*] CN=*Company Name* Service Provider CVC	C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*] CN=*Company Name* Service Provider CVC
Intended Usage	CableLabs コード検証 CA は、認可された各サービス プロバイダーに対してこの証明書を発行します。この証明書は、セキュアなソフトウェア ダウンロードのために CATV 事業者により設定されたポリシーで使用されます。	
Signed By	CableLabs コード検証 CA	tComLabs コード検証 CA 証明書
Validity Period	CableLabs ポリシーにより設定	
Modulus Length	1024, 1536, 2048	
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m]	

Organization の Company Name は Common Name の Company Name と異なる場合があります。

CVC の証明書失効リスト

CVC の証明書失効リスト（CRL）をサポートする場合に、S-MTA は不要です。

APPENDIX **A**

アラートとエラー メッセージ

この付録では、Broadband Access Center（BAC）が生成するすべてのアラート メッセージとエラー メッセージについて説明します。次のメッセージを取り上げます。

- RDU のアラート（P.A-2）
- DPE のアラート（P.A-3）
- ウォッチドッグのアラート（P.A-5）
- Network Registrar 拡張ポイントのアラート（P.A-6）

BAC のアラートは、syslog サービスを通して生成されます。syslog は、Solaris 上で情報のロギングを管理するためのクライアント / サーバ プロトコルです。BAC の syslog アラートは、ロギング サービスではありません。問題が発生した場合には通知されますが、問題の原因がいつも特定されるとは限りません。この情報は、該当する BAC ログ ファイルに書き込まれる場合もあります。

メッセージ形式

BAC がアラート メッセージを生成するときの形式は次のとおりです。

XXX-#-####: Message

- *XXX*：ファシリティ コードを表します。これには、次のものが含まれます。
  - RDU（Regional Distribution Unit）
  - DPE（Device Provisioning Engine）
  - AGENT（rduSnmpAgent または dpeSnmpAgent）
  - NR_EP（Cisco Network Registrar 拡張ポイント）
  - KDC（鍵発行局）
- *#*：使用されている重大度のレベルを表します。表 A-1 にさまざまなレベルを示します。

表 A-1　アラート メッセージの重大度レベル

重大度レベル	説明
1	アラートを示します。
2	重大なアラートを示します。
3	エラーを示します。
6	情報メッセージを示します。

- *###*：数字のエラー コードを示します。
- *Message*：アラートのテキスト（メッセージ）を表します。

RDU のアラート

表 A-2 は、RDU のアラートを示しています。

表 A-2　RDU のアラート

アラート	説明
RDU-1-101: RDU ran out of disk space	RDU サーバのストレージ パーティションの容量が不足していることを示します。このエラーが発生すると、RDU は自動的に再起動を試みますが、通常は、利用可能なストレージ容量が増加するまで同じエラーが再び発生します。一部のログ ファイルは削除または圧縮できます。 詳細については、第 14 章「サポートするツールと高度な概念」を参照してください。
RDU-1-103: RDU ran out of memory	RDU のメモリが不足していることを示します。このエラーが発生すると、RDU サーバは自動的に再起動します。
RDU-1-111: Evaluation key for technology *[technology_name]* expired	指定したテクノロジーの評価キーの期限が満了していることを示します。シスコの営業担当または TAC にお問い合せのうえ、新しいライセンス キーを入手してください。
RDU-1-115: You have used *[]* percent of available *[technology_name]* licenses.	ライセンスの総許容数のうち使用されているライセンスの数をパーセントで示します。このアラートは、ライセンスの総許容量の 80% に達すると表示されます。
RDU-1-122: DNS took *[]* seconds for lookup of address *[ip/hostname]*.Check DNS configuration and health of servers	DNS からの応答に遅延が発生しているため、BAC のパフォーマンスが低下している可能性があることを示します。このアラートは、IP アドレスのルックアップが 60 秒を上回るたびに生成されます。
RDU-2-119: Directory *[]* that contains the RDU database has a filesystem block size of *[]* bytes that does not match the required size of *[]* bytes.Corruption may occur.	データベース ファイルが含まれるファイル システムが、8 KB 以上のブロック サイズをサポートするように設定されていないため、BAC データベースが信頼できない可能性があることを示します。 ファイル システムのブロック サイズの設定の詳細については、『*Installation and Setup Guide for the Cisco Broadband Access Center 4.0*』を参照してください。
RDU-2-200: Directory *[]* that contains the RDU database transaction logs has a filesystem block size of *[]* bytes that does not match the required size of *[]* bytes.Corruption may occur.	データベース ログ ファイルが含まれるファイル システムが、8 KB 以上のブロック サイズをサポートするように設定されていないため、BAC データベースが信頼できない可能性があることを示します。 ファイル システムのブロック サイズの設定の詳細については、『*Installation and Setup Guide for the Cisco Broadband Access Center 4.0*』を参照してください。

（注） RDU の syslog アラートが送信されるたびに、追加の詳細が *BPR_DATA/rdu/logs/rdu.log* というログ ファイルに書き込まれます（追加の詳細がある場合）。

DPE のアラート

DPE の syslog アラートが送信されるたびに、追加の詳細が DPE ログに書き込まれます。

DPE ログにアクセスするには、**show log** コマンドを使用します。詳細については、『*Cisco Broadband Access Center DPE CLI Reference 4.0*』を参照してください。

DPE エラーの中には、RDU サーバのログ ファイルに伝播されるものもあります。これらのエラーは、*BPR_DATA/rdu/logs/rdu.log* ファイルで確認できます。

表 A-3 は、DPE のアラートを示しています。

表 A-3　DPE のアラート

アラート	説明
DPE-1-102: DPE ran out of disk space	DPE サーバが使用するストレージ パーティションの容量が不足しています。次の 3 つの解決策があります。 **a.** ディスクに常駐する過剰のサポート バンドルをクリアします。そのためには、不要なサポート バンドルを別のコンピュータへ移動した後、DPE コマンドライン インターフェイス（CLI）から **clear bundles** コマンドを実行します。 **b.** DPE の CLI から **clear logs** コマンドを実行して、ディスク領域をクリアします。 **c.** 最後の手段として、DPE の CLI から **clear cache** コマンドを実行して、すべてのキャッシュ ファイルを削除し、DPE を強制的に RDU サーバと再同期します。
DPE-1-104: DPE ran out of memory	DPE プロセスのメモリが不足しています。このエラー状態になると、DPE は自動的に再起動します。 DPE に存在するデバイス構成の数を確認します。デバイス構成の数が多いほど、使用されるメモリは多くなります。デバイス構成の数を減らすには、DPE がサービスするプロビジョニング グループ（プライマリまたはセカンダリ）内のデバイスの数を制限します。
DPE-1-109: Failed to connect to RDU	RDU に接続できません。次の作業を行う必要があります。 **a.** DPE ネットワークが正しく構成および接続されていることを確認します。 **b.** **dpe rdu-server** コマンドを使用して、DPE が正しい RDU に接続するよう設定されていること、および接続ポートが正しく設定されていることを確認します。 **c.** RDU プロセスが正しいサーバで実行され、正しいポートで受信されていることを確認します。RDU への接続が確立されるまで、数秒ごとに DPE から RDU プロセスへの再接続が試行されます。
DPE-1-117: DPE license nodes have been exceeded or there is no valid DPE license	DPE を開始する BAC プロセス ウォッチドッグで DPE のライセンスが検出されなかったことを示します。 管理者のユーザ インターフェイスを使用して、DPE のライセンス キーを入力します。ライセンスがない場合は、シスコ代理店にお問い合せください。

表 A-3　DPE のアラート（続き）

アラート	説明
DPE-1-116: DPE evaluation license has expired.Dropping DPE connections and deleting DPEs from database	DPE の評価ライセンス キーの期限が満了していることを示します。シスコの営業担当または TAC にお問い合せのうえ、新しいライセンス キーを入手してください。
DPE-2-118: Directory *[]* that contains the DPE's cache has a filesystem block size of *[]* bytes that does not match the required size of *[]* bytes.Corruption may occur.	ファイル システムが、8 KB 以上のブロック サイズをサポートするように設定されていないため、DPE キャッシュが信頼できない可能性があることを示します。 ファイル システムのブロック サイズの設定の詳細については、『*Installation and Setup Guide for the Cisco Broadband Access Center 4.0*』を参照してください。
DPE-1-121: Cannot start the server due to an invalid encryption key.	暗号キーが無効であるため DPE を開始できなかったことを示します。

ウォッチドッグのアラート

プロセス ウォッチドッグによって syslog アラートが送信されるたびに、エラーの詳細が *BPR_DATA/agent/logs/agent_console.log* ファイルに書き込まれます（エラーの詳細がある場合）。また、アラートで言及されている特定のコンポーネントに対応したログ ファイルにも出力されます。たとえば、*The rdu unexpectedly terminated* のようなアラートを受信した場合は、RDU サーバのログ ファイル（*BPR_DATA/rdu/logs/rdu.log*）で追加の情報を確認します。表 A-4 はプロセス ウォッチドッグのアラートを示しています。

表 A-4　プロセス ウォッチドッグのアラート

アラート	説明
AGENT-3-9001: Failed to start the *[component]*	ウォッチドッグが特定のコンポーネントの開始に失敗したことを示します。
AGENT-3-9002: The *[component]* unexpectedly terminated	プロセス ウォッチドッグで監視されていた特定のコンポーネントが、不意に失敗したことを示します。
AGENT-6-9004: The *[component]* has started	プロセス ウォッチドッグによってコンポーネントが正常に開始されるたびに生成されます。このメッセージは情報の提供のみを目的としています。
AGENT-6-9005: The *[component]* has stopped	プロセス ウォッチドッグによってコンポーネントが正常に停止されるたびに生成されます。このメッセージは情報の提供のみを目的としています。
AGENT-3-9003: Failed to stop the *[component]*	プロセス ウォッチドッグが終了しようとしたコンポーネントが停止しなかったことを示します。
AGENT-3-9003: Failed to create listener thread; *[error no]*Failed to close listen socket; *[error no]* Failed to cancel listen thread, and so on	他のアラート メッセージで定義されていないエラーを示します。

表 A-4 でプロセス ウォッチドッグのアラート リストに示されている *[component]* 変数は、次のコンポーネント値のいずれかを表します。

- rdu
- dpe
- tomcat
- cli
- snmpAgent
- kdc

Network Registrar 拡張ポイントのアラート

BAC Network Registrar 拡張ポイントの syslog アラートが送信されるたびに、追加の詳細情報が Network Registrar ログ ファイルに書き込まれます。

表 A-5 は、Network Registrar 拡張のアラートを示しています。

表 A-5　Network Registrar 拡張のアラート

アラート	説明
NR_EP-1-106: Failed to connect to RDU	Network Registrar サーバが RDU に接続できないことを示します。RDU プロセスが実行されているかどうか確認し、まだ実行されていない場合は RDU を開始します。 RDU が実行されている場合は、Network Registrar コンピュータを使用して RDU への ping を実行します。RDU への ping を実行できない場合は、2 つのデバイス間のルーティング テーブルやその他の通信パラメータを修正します。 このアラートが頻繁に繰り返される場合は、2 つのホスト間の接続が不安定になっている可能性があります。一般的なネットワークのトラブルシューティング方法を使用して、2 つのホスト間の接続を改善してください。
NR_EP-1-107: Failed to connect to any DPEs	Network Registrar 拡張が DPE に接続できないことを示します。 プロビジョニング グループの DPE が Network Registrar 拡張ごとに存在するかどうか確認します。存在しない場合は、Network Registrar プロビジョニング グループを、DPE を利用できるものに変更します。DPE がプロビジョニング グループにある場合は、Network Registrar 拡張が RDU に登録されていることを確認します。登録されていない場合、DPE は認識されません。 この確認を行った後もアラートが継続する場合は、Network Registrar 拡張とプロビジョニング グループの DPE との間のネットワーク接続を確認します。 このアラートが頻繁に繰り返される場合は、2 つのホスト間の接続が不安定になっている可能性があります。一般的なネットワークのトラブルシューティング方法を使用して、2 つのホスト間の接続を改善してください。
NR_EP-6-108: The BAC NR extensions have started	Network Registrar 拡張が起動しました。
NR_EP-6-109: The BAC NR extensions have stopped	Network Registrar 拡張が停止しました。
NR_EP-6-110: Registered with RDU *[address and port]*	Network Registrar 拡張が RDU に登録されました。*address and port* は、Network Registrar 拡張が登録された RDU のアドレスを示します。
NR_EP-1-111: Failed to find usable (best) DPEs	Network Registrar 拡張が使用可能な DPE を見つけられなかったことを示します。

APPENDIX **B**

オプションのサポート

この付録では、BAC がサポートするテクノロジー固有のオプションについて、テクノロジーのバージョンごとに説明します。また、オプションごとに次の属性を指定します。

- オプション番号：オプション番号を、整数かドット付き表記で示します。
- 説明：オプションについて説明します。
- 符号化：データ形式、およびオプション値の符号化を指定します。符号化タイプの詳細については、P.5-15 の「定義済みオプションの符号化タイプ」を参照してください。
- 検証：許容されるオプション値を制限する検証ルールを指定します。
- 多値：1 つの設定ファイルに複数のオプションを指定できるかどうかを示します。サブオプションの場合、親オプション内でサブオプションの繰り返しが可能かどうかを指定します。
- バージョン：そのオプション番号と符号化がサポートされるテクノロジー バージョンを示します。

この付録では、次のテクノロジーのオプションについて説明します。

DOCSIS オプションのサポート

表 B-1 は、DOCSIS のオプションと、各オプションに固有のバージョンのサポートを示します。

表 B-1　DOCSIS のオプションとバージョンのサポート

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
0	PAD	長さ、値ともになし	なし	あり	✓	✓	✓	✓
1	ダウンストリーム周波数	32 ビット符号なし整数	62500 の倍数	なし	✓	✓	✓	✓
2	アップストリーム チャネル ID	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
3	ネットワーク アクセス コントロール	ブール	なし	なし	✓	✓	✓	✓
4	サービス クラス	複合	なし	あり	✓	✓	✓	✓
4.1	クラス ID	8 ビット符号なし整数	1 ～ 16	なし	✓	✓	✓	✓
4.2	最大ダウンストリームレート	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
4.3	最大アップストリームレート	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
4.4	アップストリーム チャネルの優先順位	8 ビット符号なし整数	8 未満	なし	✓	✓	✓	✓
4.5	アップストリーム チャネルの保証最低速度	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
4.6	アップストリーム チャネルの最大伝送バースト	16 ビット符号なし整数	なし	なし	✓	✓	✓	✓
4.7	サービス クラス プライバシー イネーブル	ブール	なし	なし	✓	✓	✓	✓
6	CM MIC 設定内容	16 バイト	なし	なし	✓	✓	✓	✓
7	CMTS MIC 設定内容	バイト	なし	なし	✓	✓	✓	✓
9	ソフトウェア アップグレードのファイル名	NVTASCII	なし	なし	✓	✓	✓	✓
10	SNMP 書き込みアクセス コントロール	OIDCF	なし	あり	✓	✓	✓	✓
11	SNMP の MIB オブジェクト	SNMPVarBind	なし	あり	✓	✓	✓	✓
14	CPE のイーサネット MAC アドレス	MAC アドレス	なし	あり	✓	✓	✓	✓
15	電話の設定オプション	複合	なし	なし	✓	✓	✓	✓
15.2	サービス プロバイダー名	NVTASCII	なし	なし	✓	✓	✓	✓
15.3	電話番号（1）	NVTASCII	なし	なし	✓	✓	✓	✓
15.4	電話番号（2）	NVTASCII	なし	なし	✓	✓	✓	✓
15.5	電話番号（3）	NVTASCII	なし	なし	✓	✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
15.6	接続のしきい値	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
15.7	ログイン ユーザ名	NVTASCII	なし	なし	✓	✓	✓	✓
15.8	ログイン パスワード	NVTASCII	なし	なし	✓	✓	✓	✓
15.9	DHCP 認証	ブール	なし	なし	✓	✓	✓	✓
15.10	DHCP サーバ	IP アドレス	なし	なし	✓	✓	✓	✓
15.11	RADIUS レルム	NVTASCII	なし	なし	✓	✓	✓	✓
15.12	PPP 認証	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
15.13	デマンド ダイヤルの非アクティビティ タイマーのしきい値	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
16	SNMP IPv4 アドレス（廃止）	IP アドレス	なし	なし	✓	✓	✓	✓
17	ベースライン　プライバシーの設定内容	複合	なし	なし	✓	✓	✓	✓
17.1	認可待機タイムアウト	32 ビット符号なし整数	1 ～ 30	なし	✓	✓	✓	✓
17.2	再認可待機タイムアウト	32 ビット符号なし整数	1 ～ 30	なし	✓	✓	✓	✓
17.3	認可猶予時間	32 ビット符号なし整数	1 ～ 1800	なし	✓			
17.3	認可猶予時間	32 ビット符号なし整数	1 ～ 6047999	なし		✓	✓	✓
17.4	操作待機タイムアウト	32 ビット符号なし整数	1 ～ 10	なし	✓	✓	✓	✓
17.5	キー再生成待機タイムアウト	32 ビット符号なし整数	1 ～ 10	なし	✓	✓	✓	✓
17.6	TEK の猶予時間	32 ビット符号なし整数	1 ～ 1800	なし	✓			
17.6	TEK の猶予時間	32 ビット符号なし整数	1 ～ 302399	なし		✓	✓	✓
17.7	認可拒否待機タイムアウト	32 ビット符号なし整数	1 ～ 600	なし	✓	✓	✓	✓
17.8	SA マップの待機タイムアウト	32 ビット符号なし整数	1 ～ 10	なし		✓	✓	✓
17.9	SA マップの最大リトライ回数	32 ビット符号なし整数	1 ～ 10	なし		✓	✓	✓
18	CPE の最大数	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
19	TFTP サーバのタイムスタンプ	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
20	TFTP サーバのプロビジョニング済みモデムのアドレス	IP アドレス	なし	なし	✓	✓	✓	✓
21	ソフトウェア アップグレードの TFTP サーバ	IP アドレス	なし	なし	✓	✓	✓	✓
22	アップストリーム パケットの分類の符号化	複合	なし	あり		✓	✓	✓
22.1	分類子の参照	8 ビット符号なし整数	1 ～ 255	なし		✓	✓	✓
22.2	分類子の識別子	16 ビット符号なし整数	1 ～ 65535	なし		✓	✓	✓
22.3	サービス フローの参照	16 ビット符号なし整数	1 ～ 65535	なし		✓	✓	✓
22.4	サービス フローの識別子	32 ビット符号なし整数	1 以上	なし		✓	✓	✓
22.5	ルールの優先順位	8 ビット符号なし整数	なし	なし		✓	✓	✓
22.6	分類子のアクティベーション状態	ActInact	なし	なし		✓	✓	✓
22.7	動的サービスの変更アクション	8 ビット符号なし整数	3 未満	なし		✓	✓	✓
22.9	IPv4 パケットの分類の符号化	複合	なし	なし		✓	✓	✓
22.9.1	サービス範囲とマスクの IPv4 タイプ	3 ビット バイトの 8 ビット符号なし整数	なし	なし		✓	✓	✓
22.9.2	IP プロトコル	16 ビット符号なし整数	258 未満	なし		✓	✓	✓
22.9.3	IPv4 送信元アドレス	IP アドレス	なし	なし		✓	✓	✓
22.9.4	IPv4 送信元マスク	IP アドレス	なし	なし		✓	✓	✓
22.9.5	IPv4 宛先アドレス	IP アドレス	なし	なし		✓	✓	✓
22.9.6	IPv4 宛先マスク	IP アドレス	なし	なし		✓	✓	✓
22.9.7	TCP/UDP 送信元ポートの開始	16 ビット符号なし整数	なし	なし		✓	✓	✓
22.9.8	TCP/UDP 送信元ポートの終了	16 ビット符号なし整数	なし	なし		✓	✓	✓
22.9.9	TCP/UDP 宛先ポートの開始	16 ビット符号なし整数	なし	なし		✓	✓	✓
22.9.10	TCP/UDP 宛先ポートの終了	16 ビット符号なし整数	なし	なし		✓	✓	✓
22.10	イーサネット LLC パケットの分類の符号化	複合	なし	なし		✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
22.10.1	宛先 MAC アドレス	MAC アドレスとマスク	なし	なし		✓	✓	✓
22.10.2	送信元 MAC アドレス	MAC アドレス	なし	なし		✓	✓	✓
22.10.3	Ethertype/DSAP/ MacType	8 ビット符号なし整数と 16 ビット符号なし整数	なし	なし		✓	✓	✓
22.11	IEEE 802.1P/Q パケットの分類の符号化	複合	なし	なし		✓	✓	✓
22.11.1	IEEE 802.1P User_Priority	8 ビット符号なし整数のペア	8 未満	なし		✓	✓	✓
22.11.2	IEEE 802.1Q VLAN_ID	16 ビット符号なし整数	なし	なし		✓	✓	✓
22.12	IPv6 パケットの分類の符号化	複合	なし	なし				✓
22.12.1	IPv6 トラフィック クラスの範囲とマスク	3 ビット バイトの 8 ビット符号なし整数	なし	なし				✓
22.12.2	IPv6 フロー ラベル	32 ビット符号なし整数	1 以上	なし				✓
22.12.3	IPv6 ネクスト ヘッダー タイプ	16 ビット符号なし整数	258 未満	なし				✓
22.12.4	IPv6 送信元アドレス	IPv6 アドレス	なし	なし				✓
22.12.5	IPv6 送信元プレフィックスの長さ	8 ビット符号なし整数	129 未満	なし				✓
22.12.6	IPv6 宛先アドレス	IPv6 アドレス	なし	なし				✓
22.12.7	IPv6 宛先プレフィックスの長さ	8 ビット符号なし整数	129 未満	なし				✓
22.13	CM インターフェイス マスク（CMIM）	バイト	なし	なし				✓
22.43	ベンダー固有の分類子パラメータ	複合	なし	なし		✓	✓	✓
22.43.8	ベンダー ID	OUI	なし	なし		✓	✓	✓
22.43.5	L2VPN の符号化	複合	なし	なし			✓	✓
22.43.5.1	VPNID のサブタイプ	バイト	なし	なし			✓	✓
22.43.5.2	NSI カプセル化のサブタイプ	複合	なし	なし			✓	✓
22.43.5.2.1	その他の形式のサブタイプ	長さ、値ともになし	なし	なし			✓	✓
22.43.5.2.2	IEEE 802.1Q 形式のサブタイプ	16 ビット符号なし整数	なし	なし			✓	✓
22.43.5.2.3	IEEE 802.1ad 形式のサブタイプ	32 ビット符号なし整数	なし	なし			✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
22.43.5.2.4	MPLS ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
22.43.5.2.5	L2TPv3 ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
22.43.5.3	eSAFE DHCP スヌーピングのイネーブル化	バイト	なし	なし			✓	✓
22.43.5.4	CM インターフェイス マスク	バイト	なし	なし			✓	✓
22.43.5.5	添付のグループ ID	バイト	0 ～ 16	なし			✓	✓
22.43.5.6	送信元添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
22.43.5.7	宛先添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
22.43.5.8	入力ユーザの優先順位	8 ビット符号なし整数	0 ～ 7	なし			✓	✓
22.43.5.9	ユーザの優先順位の範囲	16 ビット符号なし整数	なし	なし			✓	✓
22.43.5.43	ベンダー固有	複合	なし	なし			✓	✓
23	ダウンストリーム パケットの分類の符号化	複合	なし	あり		✓	✓	✓
23.1	分類子の参照	8 ビット符号なし整数	1 ～ 255	なし		✓	✓	✓
23.2	分類子の識別子	16 ビット符号なし整数	1 ～ 65535	なし		✓	✓	✓
23.3	サービス フローの参照	16 ビット符号なし整数	1 ～ 65535	なし		✓	✓	✓
23.4	サービス フローの識別子	32 ビット符号なし整数	1 以上	なし		✓	✓	✓
23.5	ルールの優先順位	8 ビット符号なし整数	なし	なし		✓	✓	✓
23.6	分類子のアクティベーション状態	ブール	なし	なし		✓	✓	✓
23.7	動的サービスの変更アクション	8 ビット符号なし整数	3 未満	なし		✓	✓	✓
23.8	分類子エラーの符号化	複合	なし	なし		✓	✓	✓
23.9	IPv4 パケットの分類の符号化	複合	なし	なし		✓	✓	✓
23.9.1	サービス範囲とマスクの IPv4 タイプ	3 ビット バイトの 8 ビット符号なし整数	なし	なし		✓	✓	✓
23.9.2	IP プロトコル	16 ビット符号なし整数	258 未満	なし		✓	✓	✓
23.9.3	IPv4 送信元アドレス	IP アドレス	なし	なし		✓	✓	✓
23.9.4	IPv4 送信元マスク	IP アドレス	なし	なし		✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
23.9.5	IPv4 宛先アドレス	IP アドレス	なし	なし		✓	✓	✓
23.9.6	IPv4 宛先マスク	IP アドレス	なし	なし		✓	✓	✓
23.9.7	TCP/UDP 送信元ポートの開始	16 ビット符号なし整数	なし	なし		✓	✓	✓
23.9.8	TCP/UDP 送信元ポートの終了	16 ビット符号なし整数	なし	なし		✓	✓	✓
23.9.9	TCP/UDP 宛先ポートの開始	16 ビット符号なし整数	なし	なし		✓	✓	✓
23.9.10	TCP/UDP 宛先ポートの終了	16 ビット符号なし整数	なし	なし		✓	✓	✓
23.10	イーサネット LLC パケットの分類の符号化	複合	なし	なし			✓	✓
23.10.1	宛先 MAC アドレス	MAC アドレスとマスク	なし	なし		✓	✓	✓
23.10.2	送信元 MAC アドレス	MAC アドレス	なし	なし		✓	✓	✓
23.10.3	Ethertype/DSAP/MacType	8 ビット符号なし整数と 16 ビット符号なし整数	なし	なし		✓	✓	✓
23.11	IEEE 802.1P/Q パケットの分類の符号化	複合	なし	なし		✓	✓	✓
23.11.1	IEEE 802.1P User_Priority	8 ビット符号なし整数のペア	8 未満	なし		✓	✓	✓
23.11.2	IEEE 802.1Q VLAN_ID	16 ビット符号なし整数	なし	なし		✓	✓	✓
23.12	IPv6 パケットの分類の符号化	複合	なし	なし				✓
23.12.1	IPv6 トラフィック クラスの範囲とマスク	3 ビット バイトの 8 ビット符号なし整数	なし	なし				✓
23.12.2	IPv6 フロー ラベル	32 ビット符号なし整数	1 以上	なし				✓
23.12.3	IPv6 ネクスト ヘッダー タイプ	16 ビット符号なし整数	258 未満	なし				✓
23.12.4	IPv6 送信元アドレス	IPv6 アドレス	なし	なし				✓
23.12.5	IPv6 送信元プレフィックスの長さ	8 ビット符号なし整数	129 未満	なし				✓
23.12.6	IPv6 宛先アドレス	IPv6 アドレス	なし	なし				✓
23.12.7	IPv6 宛先プレフィックスの長さ	8 ビット符号なし整数	129 未満	なし				✓
23.43	ベンダー固有の分類子パラメータ	複合	なし	なし		✓	✓	✓
23.43.5	L2VPN の符号化	複合	なし	なし			✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
23.43.5.1	VPNID のサブタイプ	バイト	なし	なし			✓	✓
23.43.5.2	NSI カプセル化のサブタイプ	複合	なし	なし			✓	✓
23.43.5.2.1	その他の形式のサブタイプ	長さ、値ともになし	なし	なし			✓	✓
23.43.5.2.2	IEEE 802.1Q 形式のサブタイプ	16 ビット符号なし整数	なし	なし			✓	✓
23.43.5.2.3	IEEE 802.1ad 形式のサブタイプ	32 ビット符号なし整数	なし	なし			✓	✓
23.43.5.2.4	MPLS ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
23.43.5.2.5	L2TPv3 ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
23.43.5.3	eSAFE DHCP スヌーピングのイネーブル化	バイト	なし	なし			✓	✓
23.43.5.4	CM インターフェイス マスク	バイト	なし	なし			✓	✓
23.43.5.5	添付のグループ ID	バイト	0 ～ 16	なし			✓	✓
23.43.5.6	送信元添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
23.43.5.7	宛先添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
23.43.5.8	入力ユーザの優先順位	8 ビット符号なし整数	0 ～ 7	なし			✓	✓
23.43.5.9	ユーザの優先順位の範囲	16 ビット符号なし整数	なし	なし			✓	✓
23.43.5.43	ベンダー固有	複合	なし	なし			✓	✓
23.43.8	ベンダー ID	OUI	なし	なし		✓	✓	✓
24	アップストリーム サービス フローのスケジューリング	複合	なし	あり		✓	✓	✓
24.1	サービス フローの参照	16 ビット符号なし整数	1 以上	なし		✓	✓	✓
24.3	サービスの識別子	16 ビット符号なし整数	なし	なし		✓	✓	✓
24.4	サービス クラス名	ZTASCII	なし	なし		✓	✓	✓
24.6	QoS パラメータ セットのタイプ	ビット フラグ 8	8 未満	なし		✓	✓	✓
24.7	トラフィックの優先順位	8 ビット符号なし整数	8 未満	なし		✓	✓	✓
24.8	アップストリームの最大持続トラフィック レート	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.9	最大トラフィック バースト	32 ビット符号なし整数	なし	なし		✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート（続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
24.10	最小予約済みトラフィックレート	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.11	最小予約済みレートのパケット サイズ（仮定）	16 ビット符号なし整数	なし	なし		✓	✓	✓
24.12	アクティブな QoS パラメータのタイムアウト値	16 ビット符号なし整数	なし	なし		✓	✓	✓
24.13	公認の QoS パラメータのタイムアウト値	16 ビット符号なし整数	なし	なし		✓	✓	✓
24.14	最大連結バースト	16 ビット符号なし整数	なし	なし		✓	✓	✓
24.15	サービス フローのスケジューリングのタイプ	サービス フロー	1 ～ 6	なし		✓	✓	✓
24.16	要求 / 伝送のポリシー	ビット フラグ 32	512 未満	なし		✓	✓	✓
24.17	公称のポーリング間隔	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.18	許容ポーリング ジッタ	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.19	任意の許可サイズ	16 ビット符号なし整数	なし	なし		✓	✓	✓
24.20	公称の許可間隔	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.21	許容許可ジッタ	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.22	間隔あたりの許可数	8 ビット符号なし整数	128 未満	なし		✓	✓	✓
24.23	IPv4 サービス タイプの上書き	8 ビット符号なし整数のペア	なし	なし		✓	✓	✓
24.24	任意の許可時間の参照	32 ビット符号なし整数	なし	なし		✓	✓	✓
24.25	コンテンション要求バックオフ ウィンドウの乗数	8 ビット符号なし整数	4 ～ 12	なし				✓
24.26	要求バイト数の乗数	8 ビット符号なし整数	1、2、4、8、16 のいずれかの値	なし				✓
24.27	SID クラスタあたりの最大要求数	8 ビット符号なし整数	256 未満	なし				✓
24.28	SID クラスタあたりの最大未処理バイト数	32 ビット符号なし整数	4294967296 未満	なし				✓
24.29	SID クラスタあたりの最大合計要求バイト数	32 ビット符号なし整数	4294967296 未満	なし				✓
24.30	SID クラスタ内の最大時間	16 ビット符号なし整数	65535 未満	なし				✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
24.31	サービス フローの必須属性マスク	ビット フラグ 32	なし	なし				✓
24.32	サービス フローの禁止属性マスク	ビット フラグ 32	なし	なし				✓
24.33	サービス フローの属性集約マスク	ビット フラグ 32	なし	なし				✓
24.34	アプリケーションの識別子	ビット フラグ 32	なし	なし				✓
24.43	ベンダー固有の QoS パラメータ	複合	なし	なし		✓	✓	✓
24.43.8	ベンダー ID	OUI	なし	なし		✓	✓	✓
24.43.5	L2VPN の符号化	複合	なし	なし			✓	✓
24.43.5.1	VPNID のサブタイプ	バイト	なし	なし			✓	✓
24.43.5.2	NSI カプセル化のサブタイプ	複合	なし	なし			✓	✓
24.43.5.2.1	その他の形式のサブタイプ	長さ、値ともになし	なし	なし			✓	✓
24.43.5.2.2	IEEE 802.1Q 形式のサブタイプ	16 ビット符号なし整数	なし	なし			✓	✓
24.43.5.2.3	IEEE 802.1ad 形式のサブタイプ	32 ビット符号なし整数	なし	なし			✓	✓
24.43.5.2.4	MPLS ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
24.43.5.2.5	L2TPv3 ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
24.43.5.3	eSAFE DHCP スヌーピングのイネーブル化	バイト	なし	なし			✓	✓
24.43.5.4	CM インターフェイス マスク	バイト	なし	なし			✓	✓
24.43.5.5	添付のグループ ID	バイト	0 ～ 16	なし			✓	✓
24.43.5.6	送信元添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
24.43.5.7	宛先添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
24.43.5.8	入力ユーザの優先順位	8 ビット符号なし整数	0 ～ 7	なし			✓	✓
24.43.5.9	ユーザの優先順位の範囲	16 ビット符号なし整数	なし	なし			✓	✓
24.43.5.43	ベンダー固有	複合	なし	なし			✓	✓
25	ダウンストリーム サービス フローのスケジューリング	複合	なし	あり		✓	✓	✓
25.1	サービス フローの参照	16 ビット符号なし整数	1 以上	なし		✓	✓	✓
25.3	サービスの識別子	16 ビット符号なし整数	なし	なし		✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
25.4	サービス クラス名	ZTASCII	なし	なし		✓	✓	✓
25.6	QoS パラメータ セットのタイプ	ビット フラグ 8	8 未満	なし		✓	✓	✓
25.7	トラフィックの優先順位	8 ビット符号なし整数	8 未満	なし		✓	✓	✓
25.8	ダウンストリームの最大持続トラフィック レート	32 ビット符号なし整数	なし	なし		✓	✓	✓
25.9	最大トラフィック バースト	32 ビット符号なし整数	なし	なし		✓	✓	✓
25.10	最小予約済みトラフィックレート	32 ビット符号なし整数	なし	なし		✓	✓	✓
25.11	最小予約済みレートのパケット サイズ（仮定）	16 ビット符号なし整数	なし	なし		✓	✓	✓
25.12	アクティブな QoS パラメータのタイムアウト値	16 ビット符号なし整数	なし	なし		✓	✓	✓
25.13	公認の QoS パラメータのタイムアウト値	16 ビット符号なし整数	なし	なし		✓	✓	✓
25.14	最大ダウンストリーム遅延	32 ビット符号なし整数	なし	なし		✓	✓	✓
25.23	IPv4 サービス タイプ（DSCP）の上書き	8 ビット符号なし整数のペア	なし	なし				✓
25.31	サービス フローの必須属性マスク	ビット フラグ 32	なし	なし				✓
25.32	サービス フローの禁止属性マスク	ビット フラグ 32	なし	なし				✓
25.33	サービス フローの属性集約マスク	ビット フラグ 32	なし	なし				✓
25.34	アプリケーションの識別子	ビット フラグ 32	なし	なし				✓
25.43	ベンダー固有の QoS パラメータ	複合	なし	なし		✓	✓	✓
25.43.8	ベンダー ID	OUI	なし	なし		✓	✓	✓
26	ペイロード ヘッダー抑制	複合	なし	あり		✓	✓	✓
26.1	分類子の参照	8 ビット符号なし整数	1 以上	なし		✓	✓	✓
26.2	分類子の識別子	16 ビット符号なし整数	1 以上	なし		✓	✓	✓
26.3	サービス フローの参照	16 ビット符号なし整数	1 以上	なし		✓	✓	✓
26.4	サービス フローの識別子	32 ビット符号なし整数	1 以上	なし		✓	✓	✓
26.5	動的サービスの変更アクション	SrvChangeAct	4 未満	なし		✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
26.7	ペイロード ヘッダー抑制フィールド（PHSF）	バイト	なし	なし		✓	✓	✓
26.8	ペイロード ヘッダー抑制索引（PHSI）	8 ビット符号なし整数	1 以上	なし		✓	✓	✓
26.9	ペイロード ヘッダー抑制マスク（PHSM）	バイト	なし	なし		✓	✓	✓
26.10	ペイロード ヘッダー抑制サイズ（PHSS）	8 ビット符号なし整数	なし	なし		✓	✓	✓
26.11	ペイロード ヘッダー抑制検証（PHSV）	検証	なし	なし		✓	✓	✓
26.13	動的ボンディング変更アクション	8 ビット符号なし整数	2 未満	なし				✓
26.43	ベンダー固有の PHS パラメータ	複合	なし	なし		✓	✓	✓
26.43.8	ベンダー ID	OUI	なし	なし		✓	✓	✓
28	分類子の最大数	16 ビット符号なし整数	なし	なし		✓	✓	✓
29	プライバシー イネーブル	ブール	なし	なし		✓	✓	✓
32	製造業者の CVC	バイト	なし	なし		✓	✓	✓
33	保証人の CVC	バイト	なし	なし		✓	✓	✓
34	SnmpV3 Kickstart の値	複合	なし	なし		✓	✓	✓
34.1	SnmpV3 Kickstart のセキュリティ名	NVTASCII	なし	なし		✓	✓	✓
34.2	SnmpV3 Kickstart のマネージャ パブリック番号	バイト	なし	なし		✓	✓	✓
35	加入者管理の制御	バイト	なし	なし		✓	✓	✓
36	加入者管理の CPE IPv4 テーブル	IP アドレス N	なし	なし		✓	✓	✓
37	加入者管理のフィルタ グループ	バイト	なし	なし		✓	✓	✓
38	SNMPv3 通知受信者	複合	なし	あり		✓	✓	✓
38.1	SNMPv3 通知受信者の IPv4 アドレス	IP アドレス	なし	なし		✓	✓	✓
38.2	SNMPv3 通知受信者の UDP ポート	16 ビット符号なし整数	なし	なし		✓	✓	✓
38.3	SNMPv3　通知受信者のトラップ タイプ	SNMP トラップ タイプ	1 ～ 5	なし		✓	✓	✓
38.4	SNMPv3 通知受信者のタイムアウト値	16 ビット符号なし整数	なし	なし		✓	✓	✓
38.5	SNMPv3 通知受信者のリトライ回数	16 ビット符号なし整数	0 ～ 255	なし		✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
38.6	通知受信者のフィルタリング パラメータ	OID	なし	なし		✓	✓	✓
38.7	通知受信者のセキュリティ名	NVTASCII	なし	なし		✓	✓	✓
38.8	SNMPv3 通知受信者の IPv6 アドレス	IPv6 アドレス	なし	なし				✓
39	Enable 2.0 モード	ブール	なし	なし			✓	✓
40	Enable Test モード	ブール	なし	あり			✓	✓
41	ダウンストリーム チャネル リスト	複合	なし	あり			✓	✓
41.1	単一ダウンストリーム チャネル	複合	なし	あり			✓	✓
41.1.1	単一ダウンストリーム チャネルのタイムアウト	16 ビット符号なし整数	なし	なし			✓	✓
41.1.2	単一ダウンストリーム チャネル周波数	32 ビット符号なし整数	62500 の倍数	なし			✓	✓
41.2	ダウンストリーム周波数範囲	複合	なし	あり			✓	✓
41.2.1	ダウンストリーム周波数範囲のタイムアウト値	16 ビット符号なし整数	なし	なし			✓	✓
41.2.2	ダウンストリーム周波数範囲の開始	32 ビット符号なし整数	62500 の倍数	なし			✓	✓
41.2.3	ダウンストリーム周波数範囲の終了	32 ビット符号なし整数	62500 の倍数	なし			✓	✓
41.2.4	ダウンストリーム周波数範囲のステップ サイズ	32 ビット符号なし整数	なし	なし			✓	✓
41.3	デフォルト スキャン	16 ビット符号なし整数	なし	あり			✓	✓
42	マルチキャスト MAC アドレス	MAC アドレス	なし	あり			✓	✓
43	DOCSIS 拡張フィールド（OUI FF-FF-FF）	複合	なし	あり	✓	✓	✓	✓
43.1	CM ロード バランシングのポリシー ID	32 ビット符号なし整数	なし	なし			✓	✓
43.2	CM ロード バランシングの優先順位	32 ビット符号なし整数	なし	なし			✓	✓
43.3	CM ロード バランシングのグループ ID	32 ビット符号なし整数	なし	なし			✓	✓
43.4	CM 範囲クラス ID の拡張	16 ビット符号なし整数	なし	なし			✓	✓
43.5	L2VPN の符号化	複合	なし	なし			✓	✓
43.5.1	VPNID のサブタイプ	バイト	なし	なし			✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
43.5.2	NSI カプセル化のサブタイプ	複合	なし	なし			✓	✓
43.5.2.1	その他の形式のサブタイプ	長さ、値ともになし	なし	なし			✓	✓
43.5.2.2	IEEE 802.1Q 形式のサブタイプ	16 ビット符号なし整数	なし	なし			✓	✓
43.5.2.3	IEEE 802.1ad 形式のサブタイプ	32 ビット符号なし整数	なし	なし			✓	✓
43.5.2.4	MPLS ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
43.5.2.5	L2TPv3 ピア形式のサブタイプ	Inet アドレス ピア	なし	なし			✓	✓
43.5.3	eSAFE DHCP スヌーピングのイネーブル化	バイト	なし	なし			✓	✓
43.5.4	CM インターフェイス マスク	バイト	なし	なし			✓	✓
43.5.5	添付のグループ ID	バイト	0 ～ 16	なし			✓	✓
43.5.6	送信元添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
43.5.7	宛先添付の個別 ID	バイト	0 ～ 16	なし			✓	✓
43.5.8	入力ユーザの優先順位	8 ビット符号なし整数	0 ～ 7	なし			✓	✓
43.5.9	ユーザの優先順位の範囲	16 ビット符号なし整数	なし	なし			✓	✓
43.5.43	ベンダー固有	複合	なし	なし			✓	✓
43.6	拡張 CMTS MIC の設定内容	複合	なし	なし				✓
43.6.1	拡張 CMTS MIC の HMAC タイプ	8 ビット符号なし整数	値 1、2、43	なし				✓
43.6.2	拡張 CMTS MIC のビットマップ	バイト	なし	なし				✓
43.6.3	明示的な拡張 CMTS MIC のダイジェストのサブタイプ	バイト	なし	なし				✓
43.7	送信元アドレス検証（SAV）認可の符号化	複合	なし	なし			✓	✓
43.7.1	CMTS に設定されている SAV グループの名前	ZTASCII	1 ～ 15	なし			✓	✓
43.7.2	SAV 静的プレフィックスのサブタイプの符号化	複合	なし	なし			✓	✓
43.7.2.1	SAV 静的プレフィックスアドレスのサブタイプ	IPv4 または IPv6 アドレス	なし	なし			✓	✓
43.7.2.2	SAV 静的プレフィックスの長さのサブタイプ	ビット フラグ 8	129 未満	なし			✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
43.8	ベンダー ID	OUI	なし	なし	✓	✓	✓	✓
43.9	ケーブル モデム マスクのサブタイプの符号化	複合	なし	なし				✓
43.9.1	ケーブル モデムの必須属性マスク	ビット フラグ 32	なし	なし				✓
43.9.2	ケーブル モデムの禁止属性マスク	ビット フラグ 32	なし	なし				✓
43.10	IP マルチキャストの加入認可の符号化	サブオプション	なし	なし				✓
43.10.1	CMTS に設定されている IP マルチキャスト プロファイルの名前	NVTASCII	1 ～ 15	あり				✓
43.10.2	IP マルチキャストの加入認可の静的セッション ルールのサブタイプの符号化	複合	なし	あり				✓
43.10.2.1	ルールの優先順位	8 ビット符号なし整数	なし	なし				✓
43.10.2.2	認可アクション	認可アクション	なし	なし				✓
43.10.2.3	送信元プレフィックス アドレスのサブタイプ	IPv4 または IPv6 アドレス	なし	なし				✓
43.10.2.4	送信元プレフィックスの長さのサブタイプ	ビット フラグ 8	129 未満	なし				✓
43.10.2.5	グループ プレフィックス アドレスのサブタイプ	IPv4 または IPv6 アドレス	なし	なし				✓
43.10.2.6	グループ プレフィックスの長さのサブタイプ	ビット フラグ 8	129 未満	なし				✓
43.10.3	最大マルチキャスト セッション数の符号化	16 ビット符号なし整数	なし	なし				✓
43	DOCSIS 拡張フィールド（OUI 00-00-0C）	複合	なし	あり	✓	✓	✓	✓
43.1	静的ダウンストリーム周波数	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.2	Sync ロス タイムアウト値	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.3	ブート モニタ イメージのアップデート	NVTASCII	なし	なし	✓	✓	✓	✓
43.4	電力のバックオフ	16 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.8	ベンダー ID	OUI	なし	なし	✓	✓	✓	✓
43.9	工場出荷時のシステム イメージのアップデート	ブール	なし	なし	✓	✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
43.10	電話回線	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.11	IP 優先順位の設定	複合	なし	あり	✓	✓	✓	✓
43.11.1	IP 優先順位の値	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.11.2	レート制限	32 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.128	IOS 設定ファイル名	NVTASCII	なし	なし	✓	✓	✓	✓
43.129	コンソールがディセーブルになっていない場合の IOS 設定ファイル	NVTASCII	なし	なし	✓	✓	✓	✓
43.131	IOS CLI コマンド	NVTASCII	なし	あり	✓	✓	✓	✓
43.132	1.0 Plus のフローの符号化	複合	なし	なし	✓	✓	✓	✓
43.132.1	1.0 Plus のフロー ID	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.132.2	クラス ID	8 ビット符号なし整数	なし	なし	✓	✓	✓	✓
43.132.3	任意の許可サイズ	16 ビット符号なし整数	1 ～ 65535	なし	✓	✓	✓	✓
43.132.4	公称の許可間隔	32 ビット符号なし整数	1 ～ 65535	なし	✓	✓	✓	✓
43.132.5	間隔あたりの許可数	8 ビット符号なし整数	0 ～ 127	なし	✓	✓	✓	✓
43.132.6	組み込み音声コール数	8 ビット符号なし整数	0 ～ 127	なし	✓	✓	✓	✓
43.132.7	保留キュー長	16 ビット符号なし整数	0 ～ 4096	なし	✓	✓	✓	✓
43.132.8	均等化キュー	複合	なし	なし	✓	✓	✓	✓
43.132.8.1	輻輳廃棄のしきい値	16 ビット符号なし整数	1 ～ 4096	なし	✓	✓	✓	✓
43.132.8.2	動的会話のキュー数	16 ビット符号なし整数	16 ～ 4096	なし	✓	✓	✓	✓
43.132.8.3	予約可能な会話のキュー数	16 ビット符号なし整数	0 ～ 1000	なし	✓	✓	✓	✓
43.132.9	カスタム キューのリスト長	8 ビット符号なし整数	1 ～ 16	なし	✓	✓	✓	✓
43.132.10	ランダム検出	ブール	なし	なし	✓	✓	✓	✓
43.132.11	プライオリティ グループ	8 ビット符号なし整数	1 ～ 16	なし	✓	✓	✓	✓
43.132.12	サービス ポリシー ファイル	NVTASCII	なし	なし	✓	✓	✓	✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
43.132.13	非アクティビティ タイマー	16 ビット符号なし整数	1 ～ 10080	なし	✓	✓	✓	✓
43.132.14	COS タグ	NVTASCII	なし	なし	✓	✓	✓	✓
43.133	ダウンストリーム サブチャネル ID	8 ビット符号なし整数	0 ～ 15	なし	✓	✓	✓	✓
43.134	SU タグ	NVTASCII	なし	なし	✓	✓	✓	✓
45	ダウンストリーム非暗号化トラフィック（DUT）のフィルタリングの符号化	複合	なし	なし			✓	✓
45.1	ダウンストリーム非暗号化トラフィック（DUT）の制御	ブール	なし	なし			✓	✓
45.2	ダウンストリーム非暗号化トラフィック（DUT）の CMIM	バイト	なし	なし			✓	✓
53	SNMPv1v2c の共存構成	複合	なし	あり				✓
53.1	SNMPv1v2c　のコミュニティ名	ZTASCII	1 ～ 32	なし				✓
53.2	SNMPv1v2c の転送アドレス アクセス	複合	なし	あり				✓
53.2.1	SNMPv1v2c の転送アドレス	転送アドレスとマスク	なし	なし				✓
53.2.2	SNMPv1v2c の転送アドレス マスク	転送アドレスとマスク	なし	なし				✓
53.3	SNMPv1v2c のアクセス ビュー タイプ	アクセス ビュー タイプ	なし	なし				✓
53.4	SNMPv1v2c のアクセス ビュー名	ZTASCII	1 ～ 32	なし				✓
54	SNMPv3 のアクセス ビュー	複合	なし	あり				✓
54.1	SNMPv3 のアクセス ビュー名	ZTASCII	1 ～ 32	なし				✓
54.2	SNMPv3 のアクセス ビュー サブツリー	OID	なし	なし				✓
54.3	SNMPv3 のアクセス ビュー マスク	バイト	1 ～ 16	なし				✓
54.4	SNMPv3 のアクセス ビュー タイプ	アクセス ビュー制御	なし	なし				✓
55	SNMP CPE アクセス コントロール	CPE アクセス コントロール	なし	なし				✓
56	チャネル割り当て設定内容	複合	なし	あり				✓
56.1	送信チャネル割り当て設定内容	8 ビット符号なし整数	なし	なし				✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
56.2	受信チャネル割り当て設定内容	32 ビット符号なし整数	なし	なし				✓
58	ソフトウェア　アップグレードの IPv6 TFTP サーバ	IPv6 アドレス	なし	なし				✓
59	TFTP のプロビジョニング済みモデムの IPv6 アドレス	IPv6 アドレス	なし	なし				✓
60	アップストリーム ドロップ パケットの分類の符号化	複合	なし	あり				✓
60.1	分類子の参照	8 ビット符号なし整数	1 ～ 255	なし				✓
60.2	分類子の識別子	16 ビット符号なし整数	1 ～ 65535	なし				✓
60.5	ルールの優先順位	8 ビット符号なし整数	なし	なし				✓
60.6	分類子のアクティベーション状態	ActInact	なし	なし				✓
60.7	動的サービスの変更アクション	8 ビット符号なし整数	3 未満	なし				✓
60.9	IPv4 パケットの分類の符号化	複合	なし	なし				✓
60.9.1	サービス範囲とマスクの IPv4 タイプ	3 ビット バイトの 8 ビット符号なし整数	なし	なし				✓
60.9.2	IP プロトコル	16 ビット符号なし整数	258 未満	なし				✓
60.9.3	IPv4 送信元アドレス	IP アドレス	なし	なし				✓
60.9.4	IPv4 送信元マスク	IP アドレス	なし	なし				✓
60.9.5	IPv4 宛先アドレス	IP アドレス	なし	なし				✓
60.9.6	IPv4 宛先マスク	IP アドレス	なし	なし				✓
60.9.7	TCP/UDP 送信元ポートの開始	16 ビット符号なし整数	なし	なし				✓
60.9.8	TCP/UDP 送信元ポートの終了	16 ビット符号なし整数	なし	なし				✓
60.9.9	TCP/UDP 宛先ポートの開始	16 ビット符号なし整数	なし	なし				✓
60.9.10	TCP/UDP 宛先ポートの終了	16 ビット符号なし整数	なし	なし				✓
60.10	イーサネット LLC パケットの分類の符号化	複合	なし	なし				✓

表 B-1　DOCSIS のオプションとバージョンのサポート（続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
60.10.1	宛先 MAC アドレス	MAC アドレスとマスク	なし	なし				✓
60.10.2	送信元 MAC アドレス	MAC アドレス	なし	なし				✓
60.10.3	Ethertype/DSAP/MacType	8 ビットおよび 16 ビット符号なし整数	なし	なし				✓
60.11	IEEE 802.1P/Q パケットの分類の符号化	複合	なし	なし				✓
60.11.1	IEEE 802.1P User_Priority	8 ビット符号なし整数のペア	8 未満	なし				✓
60.11.2	IEEE 802.1Q VLAN_ID	16 ビット符号なし整数	なし	なし				✓
60.12	IPv6 パケットの分類の符号化	複合	なし	なし				✓
60.12.1	IPv6 トラフィック クラスの範囲とマスク	3 ビット バイトの 8 ビット符号なし整数	なし	なし				✓
60.12.2	IPv6 フロー ラベル	32 ビット符号なし整数	1 以上	なし				✓
60.12.3	IPv6 ネクスト ヘッダー タイプ	16 ビット符号なし整数	258 未満	なし				✓
60.12.4	IPv6 送信元アドレス	IPv6 アドレス	なし	なし				✓
60.12.5	IPv6 送信元プレフィックスの長さ	8 ビット符号なし整数	129 未満	なし				✓
60.12.6	IPv6 宛先アドレス	IPv6 アドレス	なし	なし				✓
60.12.7	IPv6 宛先プレフィックスの長さ	8 ビット符号なし整数	129 未満	なし				✓
60.13	CM インターフェイス マスク（CMIM）	バイト	なし	なし				✓
60.43	ベンダー固有の分類子パラメータ	複合	なし	なし				✓
60.43.8	ベンダー ID	OUI	なし	なし				✓
61	加入者管理の CPE IPv6 テーブル	IPv6 アドレス N	なし	なし				✓
62	アップストリーム ドロップの分類子のグループ ID	バイト	なし	なし				✓
63	加入者管理制御の最大 CPE IPv6 プレフィックス	16 ビット符号なし整数	なし	なし				✓
64	CMTS 静的マルチキャスト セッションの符号化	複合	なし	あり				✓
64.1	静的マルチキャスト　グループの符号化	IPv4 または IPv6 アドレス	なし	なし				✓

表 B-1　DOCSIS のオプションとバージョンのサポート （続き）

オプション番号	説明	符号化	検証	多値	DOCSIS のバージョン			
					1.0	1.1	2.0	3.0
64.2	静的マルチキャストの送信元の符号化	IPv4 または IPv6 アドレス	なし	なし				✓
64.3	静的マルチキャストの CMIM の符号化	バイト	なし	なし				✓
255	データ終了（EOD）マーカー	長さ、値ともになし	なし	なし	✓	✓	✓	✓

PacketCable オプションのサポート

表 B-2 に、BAC がサポートする PacketCable MTA のオプションを示します。

表 B-2　PacketCable MTA のオプション

オプション番号	説明	符号化	検証	多値	PacketCable のバージョン		
					1.0	1.1	1.5
11	SNMP の MIB オブジェクト	1 バイト長の SNMPVarBind	なし	あり	✓	✓	✓
38	SNMPv3 通知受信者	サブオプション	なし	あり	✓	✓	✓
38.1	SNMPv3 通知受信者の IP アドレス	IP アドレス	なし	なし	✓	✓	✓
38.2	SNMPv3 通知受信者の UDP ポート番号	16 ビット符号なし整数	なし	なし	✓	✓	✓
38.3	SNMPv3 通知受信者のトラップ タイプ	SNMP のトラップ タイプ	1 ～ 5	なし	✓	✓	✓
38.4	SNMPv3 通知受信者のタイムアウト値	16 ビット符号なし整数	なし	なし	✓	✓	✓
38.5	SNMPv3 通知受信者のリトライ回数	16 ビット符号なし整数	0 ～ 255	なし	✓	✓	✓
38.6	通知受信者のフィルタリング パラメータ	OID	なし	なし	✓	✓	✓
38.7	通知受信者のセキュリティ名	NVTASCII	なし	なし	✓	✓	✓
43	ベンダー固有の情報	サブオプション	なし	あり	✓	✓	✓
43.8	ベンダー ID	OUI	なし	なし	✓	✓	✓
64	SNMP の MIB オブジェクト	2 バイト長の SNMPVarBind	なし	あり	✓	✓	✓
254	テレフォニー設定ファイルの開始 / 終了	8 ビット符号なし整数	必ず 1 または 255	なし	✓	✓	✓

CableHome オプションのサポート

表 B-3 に、BAC がサポートするノンセキュア CableHome のオプションを示します。

表 B-3　CableHome のオプションとバージョンのサポート

オプション番号	説明	符号化	検証	多値	CableHome バージョン 1.0
0	PAD	長さ、値ともになし	なし	あり	✓
9	ソフトウェア アップグレードのファイル名	NVTASCII	なし	なし	✓
10	SNMP 書き込みアクセス コントロール	OIDCF	なし	あり	✓
12	モデムの IP アドレス	IP アドレス	なし	なし	✓
14	CPE のイーサネット MAC アドレス	MAC アドレス	なし	あり	✓
21	ソフトウェア アップグレードの TFTP サーバ	IP アドレス	なし	なし	✓
28	SNMP の MIB オブジェクト	SNMPVarBind	なし	あり	✓
32	製造業者の CVC	バイト	なし	なし	✓
33	保証人の CVC	バイト	なし	あり	✓
34	SnmpV3 Kickstart の値	サブオプション	なし	なし	✓
34.1	SnmpV3 Kickstart のセキュリティ名	NVTASCII	なし	なし	✓
38	SNMPv3 通知受信者	サブオプション	なし	あり	✓
38.1	SNMPv3 通知受信者の IP アドレス	IP アドレス	なし	なし	✓
38.2	SNMPv3 通知受信者の UDP ポート番号	16 ビット符号なし整数	なし	なし	✓
38.3	SNMPv3 通知受信者のトラップタイプ	SNMP のトラップ タイプ	1 ～ 5	なし	✓
38.4	SNMPv3 通知受信者のタイムアウト値	16 ビット符号なし整数	なし	なし	✓
38.5	SNMPv3 通知受信者のリトライ回数	16 ビット符号なし整数	なし	なし	✓
38.6	通知受信者のフィルタリング パラメータ	OID	なし	なし	✓
38.7	通知受信者のセキュリティ名	NVTASCII	なし	なし	✓
43	ベンダー固有の情報	サブオプション	なし	あり	✓
43.1	ベンダー ID	OUI	なし	なし	✓
53	PS MIC。PS 設定ファイルの 20 オクテット SHA-1 ハッシュ。	バイト	なし	なし	✓
255	データ終了（EOD）マーカー	長さ、値ともになし	なし	なし	✓

APPENDIX **C**

PacketCable DHCP オプションと BAC プロパティのマッピング

この付録では、PacketCable のプロビジョニングで使用される PacketCable DHCP オプションと BAC プロパティのマッピングについて説明します。次の項目を取り上げます。

- Option 122 と BAC プロパティの比較（P.C-2）
- Option 177 と BAC プロパティの比較（P.C-3）

これらのプロパティで必要最小限のセットは、インストール中に *BPR_HOME/cnr_ep/conf/cnr_ep.properties* ファイルに設定されます。このファイルは Cisco Network Registrar ホスト上にあります。*cnr_ep.properties* で定義されたプロパティのセットは、プロビジョニング グループ内のすべての PacketCable 音声テクノロジー デバイスに適用されます。他の BAC プロパティと同様に、これらのプロパティもデバイスやサービス クラスに設定できます。管理者のユーザ インターフェイスか Application Programming Interface（API; アプリケーション プログラミング インターフェイス）のいずれかを使用して、プロパティを RDU に設定すると、*cnr_ep.properties* ファイル内の対応する設定値が上書きされます。

（注） これらの重要な設定プロパティの変更については、P.14-9 の「KeyGen ツールの使用方法」を参照してください。

BAC では PacketCable DHCP Option 122（RFC 3495 および 3594 で規定）と推奨されない PacketCable DHCP Option 177 のどちらもサポートします。BAC は、Option 122 および 177 の一方または両方のコンテンツを読み込めなくても、DHCP 要求を無視しません。Option 122 および 177 のコンテンツが使用可能であれば読み込み、オプションを無視するかどうかの判断は eMTA に委ねられます。

Option 122 と 177 の両方を要求する DHCP 要求を BAC が受信したときは、Option 177 の要求を無視し、Option 122 のコンテンツだけを読み込みます。

注意 *BPR_HOME/cnr_ep/conf/cnr_ep.properties* の各プロパティには、インスタンスが 1 つずつしかありません。

Option 122 と BAC プロパティの比較

表C-1 に、RFC 3495 および RFC 3594 の Option 122 の定義に適用される BAC プロパティを示します。

表 C-1　DHCP Option 122 と BAC プロパティの比較

DHCP オプション	タイプ	BAC プロパティ名
6	IP addr	*/ccc/dns/primary*
6	IP addr	*/ccc/dns/secondary*
122.1	IP addr	*/ccc/dhcp/primary*
122.2	IP addr	*/ccc/dhcp/secondary*
122.3	FQDN	*/ccc/prov/fqdn* **（注）** Option 122.3 は BAC が自動的に設定するので、このプロパティを手動で設定しないでください。
122.4	Integer	*/ccc/kerb/auth/backoff/nomTimeout* */ccc/kerb/auth/backoff/maxTimeout* */ccc/kerb/auth/backoff/maxRetries*
122.5	Integer	*/ccc/kerb/app/backoff/nomTimeout* */ccc/kerb/app/backoff/maxTimeout* */ccc/kerb/app/backoff/maxRetries*
122.6	String	*/ccc/kerb/realm*
122.7	Boolean	*/ccc/tgt*
122.8	Integer	*/ccc/prov/timer*
122.9	Integer	*/ccc/security/ticket/invalidation*

注意　*/ccc/kerb/auth/backoff/nomTimeout*、*/ccc/kerb/auth/backoff/maxTimeout*、または */ccc/kerb/auth/backoff/maxRetries* が 1 つでも定義されている場合、この 3 つすべてが定義される必要があります。同様に、*/ccc/kerb/app/backoff/nomTimeout*、*/ccc/kerb/app/backoff/maxTimeout*、または */ccc/kerb/app/backoff/maxRetries* が 1 つでも定義されている場合、この 3 つすべてが定義される必要があります。

Option 177 と BAC プロパティの比較

PacketCable Compliance Wave 26 では、Option 177 は推奨されておらず、優先される MTA プロビジョニング オプションは Option 122 になっています。現在も Option 177 をサポートしている従来のデバイスのために、Option 177 の定義に適用される BAC プロパティを表 C-2 に示します。

表 C-2　DHCP Option 177 と BAC プロパティの比較

Option 177	タイプ	BAC プロパティ名
177.1	ip addr	*/pktcbl/dhcp/primary*
177.2	ip addr	*/pktcbl/dhcp/secondary*
177.3	fqdn	*/pktcbl/snmp/entity/fqdn*
177.4	ip addr	*/pktcbl/dns/primary*
177.5	ip addr	*/pktcbl/dns/secondary*
177.6	string	*/pktcbl/snmp/realm*
177.7	boolean	*/pktcbl/snmp/tgt*
177.8	integer	*/pktcbl/provisioning/timer*
177.10	integer	*/pktcbl/kerberos/authentication/backoff/nomTimeout* */pktcbl/kerberos/authentication/backoff/maxTimeout* */pktcbl/kerberos/authentication/backoff/maxRetries*
177.11	integer	*/pktcbl/kerberos/application/backoff/nomTimeout* */pktcbl/kerberos/application/backoff/maxTimeout* */pktcbl/kerberos/application/backoff/maxRetries*
177.12	integer	*/pktcbl/snmp/kerberos/ticket/invalidation*

APPENDIX D

プロビジョニング API の使用例

この付録では、プロビジョニング Application Programming Interface（API; アプリケーション プログラミング インターフェイス）の最も一般的な使用例を取り上げます。詳細や、個々の API コールと機能を説明するサンプルの Java コード セグメントについては、Cisco Broadband Access Center (BAC) 4.0 の API Javadoc を参照してください。

次の使用例は、デバイスまたはサービス（またはその両方）のプロビジョニングに直接関連しています。サービス クラス、DHCP 基準、およびライセンスの管理など、多くの管理操作についてはここでは取り上げません。関連する API コールの詳細については、API Javadoc を参照することをお勧めします。これらのアクティビティのほとんどは、管理者のユーザ インターフェイスを使用して実行することもできます。

この付録では、次の使用例について説明します。

API クライアントの作成方法

各使用例を参照する前に、API クライアントの作成方法について十分に理解しておく必要があります。API クライアントを作成するには、この項で説明するワークフローを使用します。

ステップ 1　Provisioning API Command Engine（PACE）への接続を作成します。

```
// The PACE connection to use throughout the example. When
// executing multiple batches in a single process, it is advisable
// to use a single PACE connection that is retrieved at the start
// of the application. When done with the connection, YOU MUST
// explicitly close the connection with the releaseConnection()
// method call.

PACEConnection connection = null;

// Connect to the Regional Distribution Unit (RDU).
//
// The parameters defined at the beginning of this class are
// used here to establish the connection. Connections are
// maintained until releaseConnection() is called. If
// multiple calls to getInstance() are called with the same
// arguments, you must still call releaseConnection() on each
// connection you received.
//
// The call can fail for one of the following reasons:
// - The hostname / port is incorrect.
// - The authentication credentials are invalid.

try
{
    connection = PACEConnectionFactory.getInstance(

         // RDU host      rduHost,
         // RDU port      rduPort,
         // User name     userName,
         // Password       password);
}
catch (PACEConnectionException e)
{
// Handle connection error:

System.out.println("Failed to establish a PACEConnection to [" +
    userName + "@" + rduHost + ":" + rduPort + "]; " +
    e.getMessage());

System.exit(1);
}
```

ステップ 2　バッチの新しいインスタンスを作成します。

```
// To perform any operations in the Provisioning API, you must
// first create a batch. As you add commands to the batch,
// nothing gets executed until you post the batch.
// Multiple batches can be started concurrently against a
// single connection to the RDU.

Batch myBatch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);
```

ステップ 3　API コマンドをバッチに登録します。このステップの例では、*getDetails(...)* コールを使用しています。

```
// Use the Provisioning API to get all of the information for
// the specified MAC address. Since methods aren't actually
// executed until the batch is posted, the results are not
// returned until after post() completes. The getCommandStatus()
// followed by getData() calls must be used to access the results
// once the batch is posted.

final DeviceID modemMACAddress = DeviceID.getInstance("1,6,00:11:22:33:44:55",
    KeyType.MAC_ADDRESS);

List options = new ArrayList();
options.add(DeviceDetailsOption.INCLUDE_LEASE_INFO);

myBatch.getDetails(modemMACAddress, options);
```

ステップ 4　バッチを Regional Distribution Unit（RDU）サーバに送信します。

```
// Executes the batch against the RDU. All of the
// methods are executed in the order entered.

BatchStatus bStatus = null;
try
{
// Post batch in synchronous fashion without a timeout. This method will block
until
// results are returned. Other API calls are available to submit a batch with
timeout // or in asynchronous (non-blocking) fashion.

bStatus = myBatch.post();
}
catch (ProvisioningException pe)
{
System.out.println("Failed to query for modem with MAC address [" +
    modemMACAddress + "]; " + pe.getMessage());

System.exit(2);
}
```

ステップ 5　バッチのステータスを確認します。

```
// Check if any errors occurred during the execution of the
// batch. Exceptions occur during post() for truly exceptional
// situations such as failure of connectivity to RDU.
// Batch errors occur for inconsistencies such as no lease
// information for a device requiring activation. Command
// errors occur when a particular method has problems, such as
// trying to add a device that already exists.

//check batchStatus and commandStatus
//for any error

CommandStatus commandStatus = null;
if (batchStatus.getCommandCount() > 0)
{
     commandStatus = batchStatus.getCommandStatus(0);
}
if (batchStatus.isError()
    || commandStatus == null
    || commandStatus.isError())
{
        System.out.println("Failed to query for modem with MAC address [" +
           modemMACAddress + "]; " + bs.getStatusCode().toString() + ", "
           + bs.getErrorMessage());
        for (int i = 0; i < bs.getCommandCount(); i++)
        {
           CommandStatus cs = bs.getCommandStatus(i);
           System.out.println("Cmd " + i + ": status code "
                   + cs.getStatusCode().toString() + ", " + cs.getErrorMessage());
        }
}
```

エラーがなければ、バッチ コールは正常終了の結果を返します。

```
// Successfully queried for device.

System.out.println("Queried for DOCSIS modem with MAC address ["+
    modemMACAddress + "]");

// Display the results of the command (TreeMap is sorted). The
// data returned from the batch call is stored on a per-command
// basis. In this example, there is only one command, but if
// you had multiple commands all possibly returning results, you
// could access each result by the index of when it was added.
// The first method added is always index 0. From the status of
// each command, you can then access the accompanying data by
// using the getData() call. Since methods can return data of
// different types, you will have to cast the response to the
// type indicated in the Provisioning API documentation.

Map deviceData = (Map)bStatus.getCommandStatus(0).getData();

// Created a sorted map view

Map<String, Object> deviceDetails = new TreeMap(deviceData);
    for(String key: deviceDetails.keySet())
{
    System.out.println(" " + key + "=" + deviceDetails.get(key));
}
```

ステップ 6　接続を解放します。

```
// Once the last batch has been executed, the connection can
// be closed to the RDU. It is important to explicitly
// close connections since it helps ensure clean shutdown of
// the Java virtual machine.

connection.releaseConnection();
```

使用例

この項には、次の使用例を記載しています。

固定標準モードでセルフプロビジョニングされたモデムとコンピュータ

加入者は、戸建住宅内にコンピュータを設置し、DOCSIS ケーブル モデムを購入しました。コンピュータには Web ブラウザがインストールされています。

目的

次のワークフローを使用して、プロビジョニングされていない新しい DOCSIS ケーブル モデムとコンピュータをオンラインにし、適切なレベルのサービスを受けられるようにします。

ステップ 1　加入者は DOCSIS ケーブル モデムを購入して住宅内に設置し、コンピュータをケーブル モデムに接続します。

ステップ 2　加入者はモデムとコンピュータの電源をオンにし、BAC が制限付きアクセス権をモデムに付与します。コンピュータとモデムには、制限付きアクセス プールから IP アドレスが割り当てられます。

ステップ 3　加入者は Web ブラウザを起動します。スプーフィング DNS サーバにより、Web ブラウザがサービス プロバイダーの登録サーバ（OSS ユーザ インターフェイスやメディエータなど）にアクセスします。

ステップ 4　加入者はサービス プロバイダーのユーザ インターフェイスを使用して、サービス クラスの選択など、登録に必要な手順を終了します。

ステップ 5　サービス プロバイダーのユーザ インターフェイスは、加入者の情報（選択されたサービス クラスやコンピュータの IP アドレスなど）を BAC に渡します。次に、BAC が加入者のモデムとコンピュータを登録します。

```
// First we query the computer's information to find the
// modem's MAC Address. We use the computer IP Address (the web browser
// received this when the subscriber opened the service provider’s
// web interface

PACEConnection connection = PACEConnectionFactory.getInstance(
            "localhost", 49187, "admin", "changeme");

// NO_ACTIVATION is the activation mode because this is a query.
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device.
// NO_PUBLISHING is the publishing mode because we are not attempting
// to publish to external database.

Batch batch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// register getAllForIPAddress to the batch

batch.getAllForIPAddress("10.0.14.38");

BatchStatus batchStatus = null;

// post the batch to RDU server
```

```
try
{
batchStatus = batch.post();
}
catch(ProvisioningException e)
{
e.printStackTrace();
}
// Get the LeaseResults object after posting a batch.

CommandStatus commandStatus = batchStatus.getCommandStatus(0);

LeaseResults computerLease = (LeaseResults)commandStatus.getData();

// Derive the modem MAC address from computer's network
// information. The "1,6" is a standard prefix for an Ethernet
// device. The fully qualify MAC Address is required by BACC

StringBuffer modemMACAddress = new StringBuffer();
modemMACAddress.append("1,6,");
modemMACAddress.append(computerLease.getSingleLease().get("relay-agent-remote-id"))
;

// Create MacAddress object from the string

MACAddress modemMACAddressObject = new MACAddress(modemMACAddress.toString());

List<DeviceID> modemDeviceIDList = new ArrayList<DeviceID>();
    modemDeviceIDList.add(modemMACAddressObject);

// Create a new batch to add modem device

batch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// Register add API to the batch

batch.add(DeviceType.DOCSIS, modemDeviceIDList,
    null, null, "0123-45-6789", "silver", "provisioned-cm", null);

// post the batch to RDU server

// Derive computer MAC address from computer's network information.

String computerMACAddress =
    (String)computerLease.getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);

// Create a map for computer property.

Map<String, Object> properties = new HashMap<String, Object>();
properties.put(IPDeviceKeys.MUST_BE_BEHIND_DEVICE, modemMACAddress.toString());

List<DeviceID> compDeviceIDList = new ArrayList<DeviceID>();
MACAddress computerMACAddressObject = new MACAddress(computerMACAddress);
compDeviceIDList.add(computerMACAddressObject);

// Register add API to the batch
```

```
    batch.add(DeviceType.COMPUTER, compDeviceIDList,
        null, null, "0123-45-6789", null, "provisioned-cpe", properties);
    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
```

ステップ 6　プロビジョニング クライアントは *performOperation(DeviceOperation deviceOperation, DeviceID deviceID, Map<String,Object> parameters* を呼び出してモデムをリブートし、プロビジョニングされたアクセス権をモデムに付与します。

```
    // Reset the computer
    // Create a new batch

    batch = connection.newBatch(

        // No reset

        ActivationMode.AUTOMATIC,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION);

    // Register performOperation command to the batch

    batch.performOperation(DeviceOperation.RESET, modemMACAddressObject, null);

    // Post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
        }
    }
}
```

ステップ 7　ユーザ インターフェイスは、加入者にコンピュータをリブートするように求めます。

リブート後、コンピュータは新しい IP アドレスを受信します。これで、ケーブル モデムとコンピュータの両方が、プロビジョニングされたデバイスとなります。コンピュータは、サービス プロバイダーのネットワークを介してインターネットにアクセスできます。

固定標準モードでの新しいコンピュータの追加

Mmultiple System Operator（MSO; マルチプル システム オペレータ）は、加入者に対し、ケーブル モデムの背後に 2 台のコンピュータを接続することを許可しています。加入者は 1 台のコンピュータをすでに登録しています。今度は、職場から自宅にラップトップ コンピュータを持ち帰ってアクセスを設定しようとしています。加入者はハブを設置し、ラップトップをハブに接続します。

目的

次のワークフローを使用して、プロビジョニングされていない新しいコンピュータを、すでにプロビジョニングされているケーブル モデムを使用してオンラインにし、新しいコンピュータが適切なレベルのサービスを受けられるようにします。

ステップ 1　加入者は新しいコンピュータの電源をオンにし、BAC が制限付きアクセス権をコンピュータに付与します。

ステップ 2　加入者は新しいコンピュータの Web ブラウザを起動します。スプーフィング DNS サーバにより、Web ブラウザがサービス プロバイダーの登録サーバ（OSS ユーザ インターフェイスやメディエータなど）にアクセスします。

ステップ 3　加入者はサービス プロバイダーのユーザ インターフェイスを使用して、新しいコンピュータの追加に必要な手順を終了します。

ステップ 4　サービス プロバイダーのユーザ インターフェイスは、加入者の情報（選択されたサービス クラスやコンピュータの IP アドレスなど）を BAC に渡します。次に、BAC が加入者のモデムとコンピュータを登録します。

```
// First we query the computer's information to find the
// modem's MAC Address. We use the computer IP address (the web browser
// received this when the subscriber opened the service provider’s
// web interface.
PACEConnection connection = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "changeme");

// NO_ACTIVATION is the activation mode because this is a query
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device
// NO_PUBLISHING is the publishing mode because we are not attempting
// to publish to external database.

Batch batch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// register getAllForIPAddress to the batch

batch.getAllForIPAddress("10.0.14.39");
BatchStatus batchStatus = null;

// post the batch to RDU server
```

```
try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
}
// Get the LeaseResults object after posting a batch.

CommandStatus commandStatus = batchStatus.getCommandStatus(0);

LeaseResults computerLease = (LeaseResults)commandStatus.getData();

// derive the modem MAC address from computer's network
// information. The "1,6" is a standard prefix for an Ethernet
// device. The fully qualify MAC Address is required by BACC

StringBuffer modemMACAddress = new StringBuffer();
modemMACAddress.append("1,6,");
modemMACAddress.append(computerLease.getSingleLease().get("relay-agent-remote-id"))
;

// derive computer MAC address from computer's network information.

String computerMACAddress =
    (String)computerLease.getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);

//Create a map for computer property.

Map<String, Object> properties = new HashMap<String, Object>();

// setting IPDeviceKeys.MUST_BE_BEHIND_DEVICE on the computer ensures
// that when the computer boots, it will only receive its provisioned
// access when it is behind the given device. If it is not behind
// the given device, it will receive default access (unprovisioned)
// and hence fixed mode.

properties.put(IPDeviceKeys.MUST_BE_BEHIND_DEVICE, modemMACAddress);

// the IPDeviceKeys.MUST_BE_IN_PROV_GROUP ensures that the computer
// will receive its provisioned access only when it is brought up in
// the specified provisioning group. This prevents the computer
// (and/or) the modem from moving from one locality to another
// locality.

properties.put(IPDeviceKeys.MUST_BE_IN_PROV_GROUP, "bostonProvGroup");

List<DeviceID> compDeviceIDList = new ArrayList<DeviceID>();
MACAddress computerMACAddressObject = new MACAddress(computerMACAddress);
compDeviceIDList.add(computerMACAddressObject);

batch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// register add API to the batch
```

```
    batch.add(
        DeviceType.COMPUTER,    // deviceType: Computer
        compDeviceIDList,       // compDeviceIDList: the list of DeviceIDs derived from
                                // computerLease
        null,                   // hostName: not used in this example
        null,                   // domainName: not used in this example
        "0123-45-6789",         // ownerName
        null,                   // class of service: get the default COS
        "provisionedCPE",       // dhpcCriteria: Network Registrar uses this to
                                  // select a modem lease granting provisioned IP
        address properties                // device properties
    );


    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
        }
    }
}
```

ステップ 5　ユーザ インターフェイスは、加入者にコンピュータをリブートするように求めます。リブートすると、BAC が登録済みのサービス レベルをコンピュータに付与します。

これで、コンピュータはプロビジョニングされたデバイスとなり、適切なレベルのサービスにアクセスできます。

加入者のディセーブル化

加入者が滞納を繰り返した場合、サービス プロバイダーは加入者のインターネット アクセスをディセーブルにする必要があります。

目的

次のワークフローを使用して、動作中のケーブル モデムとコンピュータをディセーブルにします。その結果、デバイスがユーザのインターネット アクセスを一時的に制限します。また、この使用例では、ユーザのブラウザを特別なページにリダイレクトして、次のように通知することがあります。

```
You haven't paid your bill so your Internet access has been disabled.
```

ステップ 1　サービス プロバイダーのアプリケーションはプロビジョニング クライアント プログラムを使用して、加入者のデバイスすべてのリストを BAC に要求します。

ステップ 2　次に、サービス プロバイダーのアプリケーションはプロビジョニング クライアントを使用して、加入者のデバイスを個々にディセーブルにするか、または制限します。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

//get all for owner ID

Batch batch = conn.newBatch();
batch.getAllForOwnerID("0123-45-6789");

BatchStatus batchStatus = null;
try
{
    batchStatus = batch.post();
}
catch(Exception e)
{
    e.printStackTrace();
}
CommandStatus commandStatus = batchStatus.getCommandStatus(0);


//batch success without error, retrieve the result

RecordSearchResults rcSearchResult = (RecordSearchResults)commandStatus.getData();
List<RecordData> resultList = rcSearchResult.getRecordData();

if (resultList != null)
{
// getting the data

for (int i=0; i<resultList.size(); i++)
{
    RecordData rd = resultList.get(i);
    Map<String, Object> detailMap = rd.getDetails();

    //get the deviceType from the detail map

    String deviceType =
        (String)detailMap.get(DeviceDetailsKeys.DEVICE_TYPE);

    Key primaryKey = rd.getPrimaryKey();

    //only interest in DOCSIS
    if (DeviceType.getDeviceType(deviceType)
        .equals(DeviceType.DOCSIS))
    {

    //change COS

    batch = conn.newBatch();
    batch.changeClassOfService((DeviceID)primaryKey, "DisabledCOS");

    //change DHCPCriteria

    batch.changeDHCPCriteria((DeviceID)primaryKey, "DisabledDHCPCriteria");

    batchStatus = null;
    try
    {
         batchStatus = batch.post();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
```

```
    }
    //disable computer

    else if (DeviceType.getDeviceType(deviceType)
        .equals(DeviceType.COMPUTER))
    {
        //change DHCPCriteria

        batch = conn.newBatch();
        batch.changeClassOfService((DeviceID)primaryKey,
            "DisabledComputerCOS");
        batch.changeDHCPCriteria((DeviceID)primaryKey,
            "DisabledComputerDHCPCriteria");

        batchStatus = null;
        try
        {
            batchStatus = batch.post();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

（注） DisabledCOS の特性を定義し、モデムをリセットするときに、モデムの背後の CPE に及ぼす影響を考慮しなければならないことがあります。これは、モデムの背後に音声エンドポイントが接続されている場合に特に重要です。ケーブル モデムの動作を中断すると、その時点で進行中の通話に影響を及ぼす可能性があるためです。

これで、加入者がディセーブルになります。

モデムおよびセルフプロビジョニングされたコンピュータの事前プロビジョニング

新しい加入者はサービス プロバイダーに連絡し、サービスを要求します。加入者は、戸建住宅内にコンピュータを設置しています。サービス プロバイダーは、すべてのケーブル モデムをまとめて事前プロビジョニングします。

目的

次のワークフローを使用して、事前プロビジョニングされたケーブル モデムとプロビジョニングされていないコンピュータを、ローミング標準モードでオンラインにします。この手順は、両方のデバイスが適切なレベルのサービスを受け、登録されるために必要です。

ステップ 1　サービス　プロバイダーは、課金システムで使用される加入者のユーザ名とパスワードを選択します。

ステップ 2　サービス プロバイダーは、加入者がアクセスできるサービスを選択します。

ステップ 3　サービス プロバイダーの現場技術者は、新しい加入者の住宅内に物理ケーブルを敷設し、事前プロビジョニングされたデバイスを設置して、加入者のコンピュータに接続します。

ステップ 4　モデムの電源をオンにすると、BAC はプロビジョニングされた IP アドレスをモデムに付与します。

ステップ 5　コンピュータの電源をオンにすると、BAC はプライベート IP アドレスをコンピュータに付与します。

ステップ 6　コンピュータのブラウザ　アプリケーションを起動すると、ブラウザはサービス　プロバイダーのユーザ インターフェイスにアクセスします。

ステップ 7　サービス　プロバイダーのユーザ　インターフェイスにアクセスすると、プロビジョニングされたケーブル モデムの背後にあるコンピュータの登録に必要な手順を終了します。

```
// First we query the computer's information to find the
// modem's MAC Address. We use the computer IP address (the web browser
// received this when the subscriber opened the service provider’s
// web interface

PACEConnection connection = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "changeme");

// NO_ACTIVATION is the activation mode because this is a query
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device
// NO_PUBLISHING is the publishing mode because we are not attempting
// to publish to external database.

Batch batch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// register getAllForIPAddress to the batch

batch.getAllForIPAddress("10.0.14.38");

BatchStatus batchStatus = null;

// post the batch to RDU server

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
}

// Get the LeaseResults object after posting a batch.

CommandStatus commandStatus = batchStatus.getCommandStatus(0);

LeaseResults computerLease = (LeaseResults)commandStatus.getData();

// derive computer MAC address from computer's network information.
String computerMACAddress =
    (String)computerLease.getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);
```

```
        List<DeviceID> compDeviceIDList = new ArrayList<DeviceID>();
        MACAddress computerMACAddressObject = new MACAddress(computerMACAddress);
        compDeviceIDList.add(computerMACAddressObject);


    // NO_ACTIVATION will generate new configuration for the computer,
    // however it will not attempt to reset it.
    // NO_CONFIRMATION is the confirmation mode because we are not
    // attempting to reset the computer because this cannot be done.

    batch = connection.newBatch(
        // No reset

        ActivationMode.NO_ACTIVATION,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION,

        // No publishing to external database

        PublishingMode.NO_PUBLISHING);

    // register add API to the batch

    batch.add(
        DeviceType.COMPUTER,  // deviceType: Computer
        compDeviceIDList,     // compDeviceIDList: the list of DeviceIDs derived from
                              // computerLease
        null,                 // hostName: not used in this example
        null,                 // domainName: not used in this example
        "0123-45-6789",       // ownerName
        null,                 // class of service: get the default COS
        "provisionedCPE",     // dhpcCriteria: Network Registrar uses this to
                              // select a modem lease granting provisioned IP address
        null                  // properties: not used
    );

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

（注） *IPDeviceKeys.MUST_BE_BEHIND_DEVICE* プロパティはコンピュータに設定されませんが、このプロパティを使用すると、ケーブル モデムの背後から別のケーブル モデムへのローミングが可能になります。

ステップ 8　コンピュータを再起動すると、コンピュータはプロビジョニングされた新しい IP アドレスを受信します。

これで、ケーブル モデムとコンピュータはいずれもプロビジョニングされたデバイスとなります。コンピュータは、サービス プロバイダーのネットワークを介してインターネットにアクセスできます。

既存のモデムの修正

サービス プロバイダーの加入者は、現在 **Silver** というレベルのサービスを受けていますが、**Gold** サービスにアップグレードすることにしました。加入者は、住宅内にコンピュータを設置しています。

（注） この使用例の目的は、デバイスの修正方法を示すことです。この例は、ローミング標準以外のモードでプロビジョニングされたデバイスに適用できます。

目的

次のワークフローを使用して、既存のモデムのサービス クラスを修正し、そのサービスの変更をサービス プロバイダーの外部システムに渡します。

ステップ 1　加入者はサービス プロバイダーに電話をかけて、サービスのアップグレードを依頼します。サービス プロバイダーはユーザ インターフェイスを使用して、サービス クラスを **Silver** から **Gold** に変更します。

ステップ 2　サービス プロバイダーのアプリケーションは、BAC で次の API コールを行います。

```
    // NO_ACTIVATION is the activation mode because this is a query
    // NO_CONFIRMATION is the confirmation mode because we are not
    // attempting to reset the device
    // NO_PUBLISHING is the publishing mode because we are not attempting
    // to publish to external database.

    Batch batch = connection.newBatch(

        // No reset

        ActivationMode.NO_ACTIVATION,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION,

        // No publishing to external database

        PublishingMode.NO_PUBLISHING);

    // replace changeClassOfService to this. Make sure the comment
    // on top of this line is still there.

    batch.changeClassOfService(new MACAddress("1,6,00:11:22:33:44:55")

        // the MACAddress object

        , "Gold");


    // post the batch to the RDU

    BatchStatus batchStatus = null;
    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

これで、加入者はサービス プロバイダーのネットワークにアクセスし、*Gold* サービスを受けられるようになります。

加入者のデバイスの登録解除と削除

サービス プロバイダーは、サービスの利用を中止した加入者を削除する必要があります。

目的

次のワークフローを使用して、加入者のデバイスすべてをサービス プロバイダーのネットワークから完全に削除します。

ステップ 1　サービス プロバイダーのユーザ インターフェイスで、加入者へのサービスを停止します。

ステップ 2　次のステップでは、加入者のデバイスの登録解除方法と削除方法について説明します。サービス プロバイダーによっては、故障の場合を除いてケーブル モデムをデータベースに残す場合があるため、デバイスの削除はオプションです。また、ステップ 2-a に従ってデバイスを登録解除した場合は、ステップ 2-b に従ってデバイスを削除することはできません。

a. デバイスを登録解除するには、サービス プロバイダーのアプリケーションがプロビジョニング クライアント プログラムを使用して、加入者のデバイスすべてのリストを BAC に要求します。次に、各デバイスを登録解除してリセットします。その結果、各デバイスがデフォルトの（プロビジョニングされていない）サービス レベルに低下します。

（注）「unregister」API へのパラメータに指定されたデバイスがすでに登録解除された状態にある場合は、API コールからのステータス コードが `CommandStatusCodes.CMD_ERROR_DEVICE_UNRE- GISTER_UNREGISTERED_ERROR` に設定されます。これは通常および所定の動作です。

```
// MSO admin UI calls the provisioning API to get a list of
// all the subscriber's devices.

// Create a new connection

PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");


// NO_ACTIVATION is the activation mode because this is a query
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device
// NO_PUBLISHING is the publishing mode because we are not attempting
// to publish to external database.

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,
```

```
            // No publishing to external database

            PublishingMode.NO_PUBLISHING);

        batch.getAllForOwnerID("0123-45-6789"

        //  query all the devices for this account number
        );

        BatchStatus batchStatus = null;
        try
        {
            batchStatus = batch.post();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        CommandStatus commandStatus = batchStatus.getCommandStatus(0);


        //batch success without error, retrieve the result

        RecordSearchResults rcSearchResult = (RecordSearchResults)commandStatus.getData();
        List<RecordData> resultList = rcSearchResult.getRecordData();

        //  We need to unregister all the devices behind each modem(s) or else the
        //  unregister call for that modem will fail.

        if (resultList != null)
        {
          //Unregister the COMPUTER
          for (int i=0; i<resultList.size(); i++)
        {
            RecordData rd = resultList.get(i);
            Map<String, Object> detailMap = rd.getDetails();

            //get the deviceType from the detail map

            String deviceType = (String)detailMap.get(DeviceDetailsKeys.DEVICE_TYPE);


            //only interest in DOCSIS

            if (DeviceType.getDeviceType(deviceType) .equals(DeviceType.COMPUTER))
            {
                Key primaryKey = rd.getPrimaryKey();

                batch = conn.newBatch();
                batch.unregister((DeviceID)primaryKey);

                batchStatus = null;
                try
                {
                   batchStatus = batch.post();
                }
                catch(ProvisioningException e)
                {
                   e.printStackTrace();
                }
             }
        }
    }
        //  for each modem in the retrieved list:

        for (int i=0; i<resultList.size(); i++)
        {
        RecordData rd = resultList.get(i);
        Map<String, Object> detailMap = rd.getDetails();
```

```
    //get the deviceType from the detail map

    String deviceType = (String)detailMap.get(DeviceDetailsKeys.DEVICE_TYPE);


    //only interest in DOCSIS

    if (DeviceType.getDeviceType(deviceType) .equals(DeviceType.DOCSIS))
    {
        Key primaryKey = rd.getPrimaryKey();
        batch = conn.newBatch();
        batch.unregister((DeviceID)primaryKey);
        batchStatus = null;
        try
        {
            batchStatus = batch.post();
        }
        catch(ProvisioningException e)
        {
            e.printStackTrace();
        }
    }
}
```

b. デバイスを削除するには、サービス プロバイダーのアプリケーションがプロビジョニング クライアント プログラムを使用して、加入者の残りのデバイスをデータベースから個別に削除します。

```
    // Create a new connection

    PACEConnection conn =
        PACEConnectionFactory.getInstance("localhost", 49187, "admin", "admin123");


    // NO_ACTIVATION is the activation mode because this is a query
    // NO_CONFIRMATION is the confirmation mode because we are not
    // attempting to reset the device
    // NO_PUBLISHING is the publishing mode because we are not attempting
    // to publish to external database.

    Batch batch = conn.newBatch(

        // No reset

        ActivationMode.NO_ACTIVATION,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION,

        // No publisining to external database

        PublishingMode.NO_PUBLISHING);

        batch.getAllForOwnerID("0123-45-6789"   //  query all the devices for this
        account                                         // number
        );

        BatchStatus batchStatus = null;
        try
        {
            batchStatus = batch.post();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
```

```
        CommandStatus commandStatus = batchStatus.getCommandStatus(0);


        //batch success without error, retrieve the result

        RecordSearchResults rcSearchResult =
        (RecordSearchResults)commandStatus.getData();
        List<RecordData> resultList = rcSearchResult.getRecordData();

        if (resultList != null)
        {

        //  for each modem in the retrieved list, delete it

        for (int i=0; i<resultList.size(); i++)
        {
            RecordData rd = resultList.get(i);
            Map<String, Object> detailMap = rd.getDetails();

            //get the deviceType from the detail map

            String deviceType = (String)detailMap.get(DeviceDetailsKeys.DEVICE_TYPE);


            //only interest in DOCSIS

            if (DeviceType.getDeviceType(deviceType) .equals(DeviceType.DOCSIS))
            {
                 Key primaryKey = rd.getPrimaryKey();

                 //change COS

                 batch = conn.newBatch();
                 batch.delete((DeviceID)primaryKey, true);

                 batchStatus = null;
                 try
                {
                     batchStatus = batch.post();
                }
                 catch(ProvisioningException e)
                {
                     e.printStackTrace();
                 }
            }
    }
}
```

無差別モードでの最初のアクティベーションのセルフプロビジョニング

加入者は、戸建住宅内にコンピュータ（ブラウザ アプリケーションがインストール済み）を設置し、DOCSIS ケーブル モデムを購入しました。

目的

次のワークフローを使用して、プロビジョニングされていない新しい DOCSIS ケーブル モデムとコンピュータをオンラインにし、適切なレベルのサービスを受けられるようにします。

ステップ 1　加入者は DOCSIS ケーブル モデムを購入し、住宅内に設置します。

ステップ 2　加入者はモデムの電源をオンにし、BAC は制限付きアクセス権をモデムに付与します。

ステップ 3　加入者はコンピュータのブラウザ アプリケーションを起動します。スプーフィング DNS サーバにより、ブラウザがサービス プロバイダーの登録サーバ（OSS ユーザ インターフェイスやメディエータなど）にアクセスします。

ステップ 4　加入者はサービス プロバイダーのユーザ インターフェイスを使用して、サービス クラスの選択など、登録に必要な手順を終了します。

サービス プロバイダーのユーザ インターフェイスは、加入者の情報（選択されたサービス クラスやコンピュータの IP アドレスなど）を BAC に渡します。加入者のケーブル モデムとコンピュータが BAC に登録されます。

ステップ 5　ユーザ インターフェイスは、加入者にコンピュータをリブートするように求めます。

```
// Create a new connection
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");


Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);


// NO_ACTIVATION is the activation mode because this is a
// query. NO_CONFIRMATION is the confirmation mode because
// we are not attempting to reset the device.
// First we query the computer's information to find the
// modem’s MAC address.
// We use the computer’s IP address (the web browser
// received this when the subscriber opened the service
// provider’s web interface).
// We also assume that "bostonProvGroup"
// is the provisioning group used in that locality.

List<String> provGroupList = new ArrayList<String>();

provGroupList.add("bostonProvGroup");


batch.getAllForIPAddress("10.0.14.38",

    // ipAddress: restricted access computer lease
    provGroupList
    // provGroups: List containing provgroup
    );

BatchStatus batchStatus = null;

// post the batch to RDU server

try
{
    batchStatus = batch.post();
}
```

```
catch(ProvisioningException e)
{
    e.printStackTrace();
}

// Get the LeaseResults object after posting a batch.

CommandStatus commandStatus = batchStatus.getCommandStatus(0);

LeaseResults computerLease = (LeaseResults)commandStatus.getData();

// Derive the modem MAC address from the computer's network
// information. The 1,6, is a standard prefix for an Ethernet
// device. The fully qualified MAC address is required by BACC

StringBuffer modemMACAddress = new StringBuffer();
modemMACAddress.append("1,6,");
modemMACAddress.append(computerLease.getSingleLease().get("relay-agent-remote-id"))
;


//create MacAddress object from the string

MACAddress modemMACAddressObject = new MACAddress(modemMACAddress.toString());

List<DeviceID> modemDeviceIDList = new ArrayList<DeviceID>();
modemDeviceIDList.add(modemMACAddressObject);


// NO_ACTIVATION is the activation mode because this is a query
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device
// NO_PUBLISHING is the publishing mode because we are not attempting
// to publish to external database.

batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

Map<String, Object> properties = new HashMap<String, Object>();

// Set the property PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED
// to enable promiscuous mode on modem

properties.put(PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED, Boolean.TRUE);

properties.put(PolicyKeys.COMPUTER_DHCP_CRITERIA, "provisionedCPE");


// enable promiscuous mode by changing the technology default

batch.changeDefaults(DeviceType.DOCSIS,
    properties, null);


// post the batch to RDU server

try
{
```

```
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
    batch = conn.newBatch(

        // No reset

        ActivationMode.NO_ACTIVATION,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION,

        // No publishing to external database

        PublishingMode.NO_PUBLISHING);


    batch.add(
        DeviceType.DOCSIS,     // deviceType: DOCSIS
        modemDeviceIDList,     // macAddress: derived from computer lease
        null,                  // hostName: not used in this example
        null,                  // domainName: not used in this example
        "0123-45-6789",        // ownerID: here, account number from billing system
        "Silver",              // ClassOfService
        "provisionedCM",       // DHCP Criteria: Network Registrar uses this to
                               // select a modem lease granting provisioned IP address
        null                   // properties:
    );
```

ステップ 6　プロビジョニング クライアントは *performOperation(…)* を呼び出してモデムをリブートし、プロビジョニングされたアクセス権をモデムに付与します。

```
    // Reset the computer
    // create a new batch
    batch = conn.newBatch(

        // No reset

        ActivationMode.AUTOMATIC,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION);

    // register performOperation command to the batch

    batch.performOperation(DeviceOperation.RESET,
        modemMACAddressObject, null);

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 7　コンピュータをリブートすると、コンピュータは新しい IP アドレスを受信します。

これで、ケーブル モデムはプロビジョニングされたデバイスとなります。コンピュータは BAC に登録されていませんが、サービス プロバイダーのネットワークを介してインターネットにアクセスできます。無差別モードのモデムの背後でオンラインになっているコンピュータは、引き続き、プロビジョニング API を使用して利用することができます。

無差別モードでの 100 台のモデムの一括プロビジョニング

サービス プロバイダーのカスタマー サービス担当者が、サービス センターで、配送する 100 台のケーブル モデムを事前プロビジョニングします。

目的

次のワークフローを使用して、すべてのモデムのモデム データを新しい加入者に配送します。カスタマー サービス担当者は、割り当て可能なモデムのリストを持っています。

ステップ 1　サービス プロバイダーの発送センターで、新しいケーブル モデムまたは再利用するケーブル モデムの MAC アドレス データがリストにまとめられます。

ステップ 2　特定のサービス センターに割り当てられたモデムが BAC に一括ロードされ、そのサービス センターの識別名が付けられます。

ステップ 3　サービス　センターでモデムを新しい加入者に配送する際、カスタマー　サービス担当者が新しいサービス パラメータを入力し、モデムの Owner ID フィールドを新しい加入者のアカウント番号に合せて変更します。

```
// Create a new connection
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");


Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);



// The activation mode for this batch should be NO_ACTIVATION.
// NO_ACTIVATION should be used in this situation because no
// network information exists for the devices because they
// have not booted yet. A configuration can’t be generated if no
// network information is present. And because the devices
// have not booted, they are not online and therefore cannot
// be reset. NO_CONFIRMATION is the confirmation mode because
// we are not attempting to reset the devices.
// Create a Map for the properties of the modem
```

```
Map properties;

// Set the property PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED to
// enable promiscuous mode on modem.
// This could be done at a system level if promiscuous mode
// is your default provisioning mode.

properties.put(PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED, Boolean.TRUE);

// The PolicyKeys.CPE_DHCP_CRITERIA is used to specify the DHCP
// Criteria to be used while selecting IP address scopes for
// CPE behind this modem in the promiscuous mode.

properties.put(PolicyKeys.COMPUTER_DHCP_CRITERIA, "provisionedCPE");

// enable promiscuous mode by changing the technology default

batch.changeDefaults(DeviceType.DOCSIS,properties, null);

BatchStatus batchStatus = null;

// post the batch to RDU server

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
        }

// for each modem MAC-address in list:

ModemLoop:
{
    batch = conn.newBatch(

       // No reset

       ActivationMode.NO_ACTIVATION,

       // No need to confirm activation

       ConfirmationMode.NO_CONFIRMATION,

       // No publishing to external database

       PublishingMode.NO_PUBLISHING);


batch.add(
    DeviceType.DOCSIS,   // deviceType: DOCSIS
    modemMACAddressList, // modemMACAddressList: the list of deviceID
    null,                // hostName: not used in this example
    null,                // domainName: not used in this example
    "0123-45-6789",      // ownerID: here, account number from billing system
    "Silver",            // ClassOfService
    "provisionedCM",     // DHCP Criteria: Network Registrar uses this to
                         // select a modem lease granting provisioned IP address
    properties           // properties:
);

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
```

```
    {
        e.printStackTrace();
    }
    // end ModemLoop.
}
```

無差別モードでの最初のアクティベーションの事前プロビジョニング

新しい加入者はサービス プロバイダーに連絡し、サービスを要求します。加入者は、戸建住宅内にコンピュータを設置しています。

目的

次のワークフローを使用して、プロビジョニングされていない新しいケーブル モデムとコンピュータをオンラインにし、適切なレベルのサービスを受けられるようにします。

ステップ 1　サービス　プロバイダーは、課金システムで使用される加入者のユーザ名とパスワードを選択します。

ステップ 2　サービス プロバイダーは、加入者がアクセスできるサービスを選択します。

ステップ 3　サービス プロバイダーは、独自のユーザ インターフェイスを使用して、デバイスを登録します。

ステップ 4　サービス プロバイダーのユーザ インターフェイスは、モデムの MAC アドレスやサービス クラスなどの情報を BAC に渡します。また、モデムが CPE DHCP 基準設定を取得します。この基準設定により、Network Registrar はモデムの背後に接続するコンピュータに対して、プロビジョニングされたアドレスを選択できます。次に、新しいモデムが BAC に登録されます。

ステップ 5　サービス プロバイダーの現場技術者は、新しい加入者の住宅内に物理ケーブルを敷設し、事前プロビジョニングされたデバイスを設置して、加入者のコンピュータに接続します。

```
//  MSO admin UI calls the provisioning API to pre-provision
//  an HSD modem.

// Create a new connection
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");


Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);


// The activation mode for this batch should be NO_ACTIVATION.
// NO_ACTIVATION should be used in this situation because no
// network information exists for the modem because it has not
```

```
// booted. A configuration cannot be generated if no network
// information is present. And because the modem has not booted,
// it is not online and therefore cannot be reset.
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the modem.
// Create a map for the properties of the modem.


Map<String, Object> properties = new HashMap<String, Object>();

// Set the property PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED
// to enable promiscuous mode on modem

properties.put(PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED, Boolean.TRUE);

properties.put(PolicyKeys.COMPUTER_DHCP_CRITERIA, "provisionedCPE");


// enable promiscuous mode by changing the technology default

batch.changeDefaults(DeviceType.DOCSIS, properties, null);

BatchStatus batchStatus = null;

// post the batch to RDU server

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
}
batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);


MACAddress macAddressObject = new MACAddress("1,6,00:11:22:33:44:55");
List<DeviceID> modemDeviceIDList = new ArrayList<DeviceID>();
modemDeviceIDList.add(macAddressObject);

batch.add(
    DeviceType.DOCSIS,      // deviceType: DOCSIS
    modemDeviceIDList,      // macAddress: derived from computer lease
    null,                   // hostName: not used in this example
    null,                   // domainName: not used in this example
    "0123-45-6789",      // ownerID: here, account number from billing system
    "Silver",            // ClassOfService
    "provisionedCM",     // DHCP Criteria: Network Registrar uses this to
                         // select a modem lease granting provisioned IP address
    null                 // properties:
);


// post the batch to RDU server

try
```

```
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 6　ケーブル モデムの電源をオンにすると、BAC はプロビジョニングされたアクセス権をケーブル モデムに付与します。

ステップ 7　コンピュータの電源をオンにすると、BAC はプロビジョニングされたアクセス権をコンピュータに付与します。

これで、ケーブル モデムとコンピュータはいずれもプロビジョニングされたデバイスとなります。コンピュータは、サービス プロバイダーのネットワークを介してインターネットにアクセスできます。

既存のモデムの交換

サービス プロバイダーは、故障したモデムを交換します。

（注）　モデムから別のモデムへのローミングを制限するオプションがコンピュータに設定されている場合、モデムを交換したときは、コンピュータに設定されているモデムの MAC アドレスも変更する必要があります。

目的

次のワークフローを使用して、加入者に提供するサービスのレベルを変更することなく、既存のケーブル モデムを新しいモデムに物理的に交換します。

ステップ 1　サービス プロバイダーは、既存のモデムの MAC アドレスを新しいモデムの MAC アドレスに変更します。

```
// Create a new connection
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database
```

```
            PublishingMode.NO_PUBLISHING);


        MACAddress macAddressObject = new MACAddress("1,6,00:11:22:33:44:55");
        List<DeviceID> modemDeviceIDList = new ArrayList<DeviceID>();
        modemDeviceIDList.add(macAddressObject);

        //  old macAddress: unique identifier for the old modem
        MACAddress oldMacAddress = new MACAddress("1,6,00:11:22:33:44:55");

        // new macAddress: unique identifier for the new modem
        MACAddress newMacAddress = new MACAddress("1,6,00:11:22:33:44:66");
        List<DeviceID> newDeviceIDs = new ArrayList<DeviceID>();
        newDeviceIDs.add(newMacAddress);

        batch.changeDeviceID(oldMacAddress, newDeviceIDs);

        // post the batch to RDU server

        try
        {
            batchStatus = batch.post();
        }
        catch(ProvisioningException e)
        {
            e.printStackTrace();
        }
    }
```

ステップ 2　サービス プロバイダーは、ケーブル モデムを交換し、その電源をオンにします。

ステップ 3　コンピュータの電源もオンにする必要があります。

これで、ケーブル モデムは完全にプロビジョニングされたデバイスとなり、適切なレベルのサービスを受けられるようになります。ケーブル モデムの背後にあるコンピュータも同様です。

無差別モードでの 2 台目のコンピュータの追加

加入者は、設置されているケーブル モデムの背後に 2 台目のコンピュータを接続します。この例では、プロビジョニング API の呼び出しは必要ありません。

目的

次のワークフローを使用して、加入者が選択したサービスが複数の CPE セットの接続を許可すること、および接続された両方のコンピュータから加入者がネットワークにアクセスできることを確認します。

ステップ 1　加入者はケーブル モデムの背後に 2 台目のコンピュータを接続します。

ステップ 2　加入者はコンピュータの電源をオンにします。

加入者が選択したサービスが複数の CPE セットの接続を許可している場合、BAC はインターネットへのアクセス権を 2 台目のコンピュータに付与します。

NAT を使用した最初のアクティベーションのセルフプロビジョニング

大学で、Network Address Translation（NAT; ネットワークアドレス変換）および DHCP 機能を持つ DOCSIS ケーブル モデムが購入されました。建物の利用者 5 人はそれぞれ、ブラウザ アプリケーションを持つコンピュータを設置しています。

目的

次のワークフローを使用して、プロビジョニングされていない新しいケーブル モデム（NAT を持つ）と、モデムの背後にあるコンピュータをオンラインにし、適切なレベルのサービスを受けられるようにします。

ステップ 1　加入者は NAT および DHCP 機能を持つケーブル モデムを購入し、集合住宅内に設置します。

ステップ 2　加入者はモデムの電源をオンにし、BAC は制限付きアクセス権をモデムに付与します。

ステップ 3　加入者はラップトップ コンピュータをケーブル モデムに接続し、モデム内の DHCP サーバは IP アドレスをラップトップに付与します。

ステップ 4　加入者はコンピュータのブラウザ アプリケーションを起動します。スプーフィング DNS サーバにより、ブラウザがサービス プロバイダーの登録サーバ（OSS ユーザ インターフェイスやメディエータなど）にアクセスします。

ステップ 5　加入者はサービス プロバイダーのユーザ インターフェイスを使用して、モデムのケーブル モデムの登録に必要な手順を終了します。登録用のユーザ インターフェイスは、モデムで NAT が使用されていることを検出し、モデムを登録して、モデムが NAT 互換のサービス クラスを受けられることを確認します。詳細については、P.D-6 の「固定標準モードでセルフプロビジョニングされたモデムとコンピュータ」を参照してください。

（注）　NAT を持つ特定のケーブル モデムでは、新しいサービス クラス設定を取得するためにコンピュータをリブートするように求められる場合があります。ケーブル モデムと NAT デバイスが別々のデバイスである場合は、NAT デバイスもコンピュータの登録と同じように登録する必要があります。

NAT を持つモデムの背後への新しいコンピュータの追加

アパートには 4 人の借家人がいて、モデムを共有し、サービス プロバイダーのネットワークにアクセスしています。このアパートの大家が新しい借家人に対し、建物のモデムを共有したインターネット アクセスを提供します。モデムには NAT および DHCP 機能が含まれています。新しい借家人は、自分のコンピュータをモデムに接続しています。

（注） この例では、プロビジョニング API の呼び出しは必要ありません。

目的

次のワークフローを使用して、プロビジョニングされていない新しいコンピュータを、すでにプロビジョニングされているケーブル モデムを使用してオンラインにし、新しいコンピュータが適切なレベルのサービスを受けられるようにします。

ステップ 1　加入者はコンピュータの電源をオンにします。

ステップ 2　これで、コンピュータはプロビジョニングされたデバイスとなり、適切なレベルのサービスにアクセスできます。

プロビジョニングされた NAT モデムは、モデムの背後にあるコンピュータをネットワーク上で非表示にします。

別の DHCP スコープへのデバイスの移動

サービス プロバイダーは、ネットワークに番号を再割り当てします。これに伴い、登録済みケーブル モデムに、別の Network Registrar スコープの IP アドレスが必要になります。

目的

プロビジョニング クライアントは DHCP 基準を変更し、ケーブル モデムは対応する DHCP スコープから IP アドレスを受信します。

ステップ 1　DOCSIS モデムの DHCP 基準を「newmodemCriteria」に変更します。

```
    // Create a new connection
    PACEConnection conn = PACEConnectionFactory.getInstance(
        "localhost", 49187, "admin", "admin123");

    Batch batch = conn.newBatch(

        // No reset

        ActivationMode.AUTOMATIC,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION,

        // No publishing to external database

        PublishingMode.NO_PUBLISHING);

    // AUTOMATIC is the Activation mode because we are attempting
    // to reset the modem so that a phone line is disabled
    // NO_CONFIRMATION is the Confirmation mode because we don't
    // want the batch to fail if we can't reset the modem.
    // This use case assumes that the DOCSIS modem has been
    // previously added to the database

    batch.changeDHCPCriteria(
        new MACAddress("1,6,ff:00:ee:11:dd:22"),  // Modem's MAC address or FQDN
              "newmodemCriteria"
    );

    // post the batch to RDU server

    BatchStatus batchStatus = null;
    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 2　モデムは、「newmodemCriteria」によって対象とされたスコープから IP アドレスを取得します。

イベントを使用したデバイス削除のロギング

サービス プロバイダーは、複数のプロビジョニング クライアントを保有しており、デバイス削除をログに記録します。

目的

どのプロビジョニング クライアントがデバイスを削除しても、プロビジョニング クライアントはイベントを 1 か所のログに記録します。

ステップ 1　デバイス削除イベントのリスナーを作成します。このクラスは、*DeviceAdapter* 抽象クラスを拡張するか、または *DeviceListener* インターフェイスを実装する必要があります。また、イベントをログに記録するために *deletedDevice(DeviceEvent ev)* メソッドを無効にする必要もあります。

```
public DeviceDeletionLogger
        extends DeviceAdapter

        //Extend the DeviceAdapter class.
{
    public void deletedDevice(DeviceEvent ev)

    //Override deletedDevice.
    {
        logDeviceDeletion(ev.getDeviceID());

        //Log the deletion.
    }
}
```

ステップ 2　*PACEConnection* インターフェイスを使用して、イベントのリスナーと修飾子を登録します。

```
DeviceDeletionLogger deviceDeletionLogger =
    new DeviceDeletionLogger();

        // Modem's MAC address or FQDN "newmodemCriteria"

DeviceEventQualifier qualifier = new DeviceEventQualifier();

// We are interested only in device deletion.

qualifier.setDeletedDevice ();

// Add device listener using PACEConnection

connection.addDeviceListener(deviceDeletionLogger, qualifier
);
```

ステップ 3　システムからデバイスが削除されると、イベントが生成され、リスナーに通知されます。

イベントを使用した RDU 接続の監視

サービス プロバイダーは 1 つのプロビジョニング クライアントを実行しており、プロビジョニング クライアントと RDU 間の接続が切断された場合に通知されるようにします。

目的

次のワークフローを使用して、接続が切断された場合にイベント インターフェイスからサービス プロバイダーに通知されるように設定します。

ステップ 1　メッセージ イベントのリスナーを作成します。このクラスは、*MessagingAdapter* 抽象クラスを拡張するか、または *MessagingListener* インターフェイスを実装する必要があります。さらに、このクラスは *connectionStopped(MessagingEvent ev)* メソッドを無効にする必要もあります。

```
// Extend the service provider's Java program using the
// provisioning client to receive Messaging events.
public MessagingNotifier
    extends MessagingAdapter

    //Extend the MessagingAdapter class.
{
    public void connectionStopped(MessagingEvent ev)

    //Override connectionStopped.
    {
        doNotification(ev.getAddress(), ev.getPort());

        //Do the notification.
    }
}
```

ステップ 2　*PACEConnection* インターフェイスを使用して、イベントのリスナーと修飾子を登録します。

```
MessagingQualifier qualifier =new MessagingQualifier();
qualifier.setConnectionDown();
MessagingNotifier messagingNotifier = new MessagingNotifier();
connnection.addMessagingListener(messagingNotifier, qualifier
);
```

ステップ 3　接続が切断されると、イベントが生成され、リスナーに通知されます。接続が中断した場合は常に、PACEConnection が自動的に RDU に再接続します。

イベントを使用したバッチ完了のロギング

サービス プロバイダーは、複数のプロビジョニング クライアントを保有しており、バッチ完了をログに記録します。

目的

どのプロビジョニング クライアントがバッチを完了しても、イベントを 1 か所のログに記録します。

ステップ 1　イベントのリスナーを作成します。このクラスは、*BatchAdapter* 抽象クラスを拡張するか、または *BatchListener* インターフェイスを実装する必要があります。また、イベントをログに記録するために *completion(BatchEvent ev)* メソッドを無効にする必要があります。

```
public BatchCompletionLogger
    extends BatchAdapter

    //Extend the BatchAdapterclass.
{
    public void completion(BatchEvent ev)
    //Override completion.
    {
        logBatchCompletion(ev.BatchStatus().getBatchID());
        //Log the completion.
    }
}
```

ステップ 2　*PACEConnection* インターフェイスを使用して、イベントのリスナーと修飾子を登録します。

```
BatchCompletionLogger batchCompletionLogger = new BatchCompletionLogger();
BatchEventQualifier qualifier = new BatchEventQualifier();
connection.addBatchListener(batchCompletionLogger , qualifier
);
```

ステップ 3　バッチが完了すると、イベントが生成され、リスナーに通知されます。

デバイスの詳細情報の取得

サービス プロバイダーは管理者に対して、特定のデバイスの詳細情報を表示することを許可します。

目的

サービス プロバイダーの管理アプリケーションは、特定のデバイスに関する既知の詳細をすべて表示します。この詳細には、MAC アドレス、リース情報、デバイスのプロビジョニング ステータス、およびデバイス タイプ（既知の場合）などがあります。

ステップ 1　管理者は、サービス プロバイダーの管理者のユーザ インターフェイスに、クエリーするデバイスの MAC アドレスを入力します。

ステップ 2　BAC は、デバイスの詳細を組み込みデータベースにクエリーします。

```
// The host name or IP address of the RDU. It is
// recommended that you normally use a fully-qualified domain name
// since it lends itself to the greatest flexibility going forward.
// For example, you could change the host running RDU without
// having to reassign IPs. For that reason, having an alias for
// the machine is better than a specific name.

final String rduHost = "localhost";

// The port number of RDU on the server.

final int rduPort = 49187;

// The user name for connecting to RDU.

final String userName = "admin";

// The password to use with the username.

final String password = "changeme";

// ------------------------------------------------------------------
// DEVICE PARAMETERS, see IPDevice.getDetails()
// ------------------------------------------------------------------

// The MAC address of the modem to be queried. MAC addresses in BAC
// must follow the simple "1,6,XX:XX:XX:XX:XX:XX" format.

final DeviceID modemMACAddress = DeviceID.getInstance("1,6,00:11:22:33:44:55",
    KeyType.MAC_ADDRESS);

// The PACE connection to use throughout the example. When
// executing multiple batches in a single process, it is advisable
// to use a single PACE connection that is retrieved at the start
// of the application. When done with the connection, YOU MUST
// explicitly close the connection with the releaseConnection()
// method call.

PACEConnection connection = null;

// 1) Connect to the Regional Distribution Unit (RDU).
//
// The parameters defined at the beginning of this class are
// used here to establish the connection. Connections are
// maintained until releaseConnection() is called. If
// multiple calls to getInstance() are called with the same
// arguments, you must still call releaseConnection() on each
// connection you received.
//
// The call can fail for one of the following reasons:
// - The hostname / port is incorrect.
// - The authentication credentials are invalid.

try
{
connection = PACEConnectionFactory.getInstance(
// RDU host   rduHost,
// RDU port   rduPort,
// User name  userName,
// Password   password
);
}
catch (PACEConnectionException e)
{
// failed to get a connection
```

```
System.out.println("Failed to establish a PACEConnection to [" +
    userName + "@" + rduHost + ":" + rduPort + "]; " +
    e.getMessage());

System.exit(1);
}
// 2) Create a new batch instance.
//
// To perform any operations in the Provisioning API, you must
// first start a batch. As you make commands against the batch,
// nothing will actually start until you post the batch.
// Multiple batches can be started concurrently against a
// single connection to the RDU.

Batch myBatch = connection.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// 3) Register the getDetails(...) with the batch.

// Use the Provisioning API to get all of the information for
// the specified MAC address. Since methods aren't actually
// executed until the batch is posted, the results are not
// returned until after post() completes. The getCommandStatus()
// followed by getData() calls must be used to access the results
// once the batch is posted.

final DeviceID modemMACAddress = DeviceID.getInstance("1,6,00:11:22:33:44:55",
        KeyType.MAC_ADDRESS);

List options = new ArrayList();
     options.add(DeviceDetailsOption.INCLUDE_LEASE_INFO);

myBatch.getDetails(modemMACAddress, options);

// 4) Post the batch to the server.
//
// Executes the batch against the RDU. All of the
// methods are executed in the order entered and the data
// changes are applied against the embedded database in RDU.

BatchStatus bStatus = null;
try
{
    bStatus = myBatch.post();
}
catch (ProvisioningException pe)
{
System.out.println("Failed to query for modem with MAC address [" +
    modemMACAddress + "]; " + pe.getMessage());

System.exit(2);
}
// 5) Check to see if the batch was successfully posted.
//
// Verify if any errors occurred during the execution of the
// batch. Exceptions occur during post() for truly exception
// situations such as failure of connectivity to RDU.
// Batch errors occur for inconsistencies such as no lease
```

```
// information for a device requiring activation. Command
// errors occur when a particular method has problems, such as
// trying to add a device that already exists.

if (bStatus.isError())
{
// Batch error occurred.

System.out.println("Failed to query for modem with MAC address [" +
    modemMACAddress + "]; " + bStatus.getErrorMessage());

System.exit(3);
}
```

ステップ 3　サービス プロバイダーのアプリケーションは、デバイス データの詳細に関するページを表示します。このページには、要求されたデバイスに関する既知の情報をすべて表示できます。デバイスがサービス プロバイダーのネットワークに接続された場合、このデータにはリース情報（IP アドレスやリレー エージェントの ID など）が含まれます。データは、デバイスがプロビジョニングされたかどうかを示します。プロビジョニングされた場合、データにはデバイス タイプも含まれます。

```
// Successfully queried for device.
System.out.println("Queried for DOCSIS modem with MAC address ["+
    modemMACAddress + "]");

// Display the results of the command (TreeMap is sorted). The
// data returned from the batch call is stored on a per-command
// basis. In this example, there is only one command, but if
// you had multiple commands all possibly returning results, you
// could access each result by the index of when it was added.
// The first method added is always index 0. From the status of
// each command, you can then access the accompanying data by
// using the getData() call. Since methods can return data of
// different types, you will have to cast the response to the
// type indicated in the Provisioning API documentation.

Map<String, Object> deviceDetails = new HashMap<String,
    Object>((Map)bStatus.getCommandStatus(0).getData());

String deviceType = (String)deviceDetails.get(DeviceDetailsKeys.DEVICE_TYPE);
String macAddress = (String)deviceDetails.get(DeviceDetailsKeys.MAC_ADDRESS);
String fqdn = (String)deviceDetails.get(DeviceDetailsKeys.FQDN);
String duid = (String)deviceDetails.get(DeviceDetailsKeys.DUID);
String host = (String)deviceDetails.get(DeviceDetailsKeys.HOST);
String domain = (String)deviceDetails.get(DeviceDetailsKeys.DOMAIN);

// if the device is DocsisModem, get the COS

String cos   = (String)deviceDetails.get(DeviceDetailsKeys.CLASS_OF_SERVICE);
String dhcpCriteria   = (String)deviceDetails.get(DeviceDetailsKeys.DHCP_CRITERIA);
String provGroup = (String)deviceDetails.get(DeviceDetailsKeys.PROV_GROUP);
Boolean isProvisioned =
(Boolean)deviceDetails.get(DeviceDetailsKeys.IS_PROVISIONED);
String ownerID = (String)deviceDetails.get(DeviceDetailsKeys.OWNER_ID);
Boolean isRegistered = (Boolean)deviceDetails.get(DeviceDetailsKeys.IS_REGISTERED);
String oidNumber =
(String)deviceDetails.get(GenericObjectKeys.OID_REVISION_NUMBER);

// if the device is a modem, get the device behind

String relayAgentMacAddress =
    (String)deviceDetails.get(DeviceDetailsKeys.RELAY_AGENT_MAC);
String relayAgentDUID =
(String)deviceDetails.get(DeviceDetailsKeys.RELAY_AGENT_DUID);

// get the map of Device property
```

```
Map deviceProperties = (Map)deviceDetails.get(DeviceDetailsKeys.PROPERTIES);

// get the map of discovery data v4

Map dhcpdiscovermapv4 =
    (Map)deviceDetails.get(DeviceDetailsKeys.DISCOVERED_DATA_DHCPV4);

// if discovery data is not null, get the inform, response, request and environment
// map from discovery data map

Map dhcpInformMap = (Map)dhcpdiscovermapv4.get("INFORM");
Map dhcpRespMap = (Map)dhcpdiscovermapv4.get("RESPONSE");
Map dhcpReqMap = (Map)dhcpdiscovermapv4.get("REQUEST");
Map dhcpEnvMap = (Map)dhcpdiscovermapv4.get("ENVIRONMENT");

// get the map of lease query v4

Map leasemapv4 = (Map)deviceDetails.get(DeviceDetailsKeys.LEASE_QUERY_DATA_DHCPV4);
String leaseTime = (String)leasemapv4.get(CNRNames.DHCP_LEASE_TIME.toString());
String rebindingTime =
    (String)leasemapv4.get(CNRNames.DHCP_REBINDING_TIME.toString());

String clientLastTransTime =
    (String)leasemapv4.get(CNRNames.CLIENT_LAST_TRANSACTION_TIME.toString());
String clientIPAddress=
(String)leasemapv4.get(CNRNames.CLIENT_IPADDRESS.toString());
String relayAgentRemoteID=
    (String)leasemapv4.get(CNRNames.RELAY_AGENT_REMOTE_ID.toString());
String relayAgentCircuitID=
    (String)leasemapv4.get(CNRNames.RELAY_AGENT_CIRCUIT_ID.toString());

// get the map of discovery DHCP v6

Map dhcpdiscovermapv6 =
    (Map)deviceDetails.get(DeviceDetailsKeys.DISCOVERED_DATA_DHCPV6);

// if discovery data is not null , get the inform, response, request and
environment
// map from discovery data map

Map dhcpv6InformMap = (Map)dhcpdiscovermapv6.get("INFORM");

Map dhcpv6RespMap = (Map)dhcpdiscovermapv6.get("RESPONSE");

Map dhcpv6ReqMap = (Map)dhcpdiscovermapv6.get("REQUEST");

Map dhcpv6RelReqMap = (Map)dhcpdiscovermapv6.get("RELAY_REQUEST");

Map dhcpv6EnvMap = (Map)dhcpdiscovermapv6.get("ENVIRONMENT");

// get the map of lease query V6

Map leasemapv6 = (Map)deviceDetails.get(DeviceDetailsKeys.LEASE_QUERY_DATA_DHCPV6);

String iaprefixkey = (String)leasemapv6.get(CNRNames.IAPREFIX.toString());
String iaaddrkey = (String)leasemapv6.get(CNRNames.IAADDR.toString());
String leasetimev6 = (String)leasemapv6.get(CNRNames.VALID_LIFETIME.toString());
String renewaltimev6 =
(String)leasemapv6.get(CNRNames.PREFERRED_LIFETIME.toString());
String dhcplasttranstimev6 =
    (String)leasemapv6.get(CNRNames.CLIENT_LAST_TRANSACTION_TIME);
String clientIpAddressv6 = (String)leasemapv6.get(CNRNames.CLIENT_IPADDRESS);
String relayagentremoteidv6 =
(String)leasemapv6.get(CNRNames.RELAY_AGENT_REMOTE_ID);
String relayagentcircuitidv6 =
    (String)leasemapv6.get(CNRNames.RELAY_AGENT_CIRCUIT_ID);
```

デバイス タイプを使用した検索

サービス プロバイダーは管理者に対して、すべての DOCSIS モデムのデータを表示することを許可します。

目的

サービス プロバイダーの管理アプリケーションは、DOCSIS デバイスのリストを返します。

ステップ 1　管理者は、サービス プロバイダーの管理者のユーザ インターフェイスで、検索オプションを選択します。

ステップ 2　BAC は、DOCSIS モデムの MAC アドレスすべてのリストを、組み込みデータベースにクエリーします。

```
public static void getAllDevicesByDeviceType() throws Exception  {
      DeviceSearchType dst = DeviceSearchType.getByDeviceType(
          DeviceType.getDeviceType(DeviceTypeValues.DOCSIS_MODEM),
          ReturnParameters.ALL);

      RecordSearchResults rs = null;

      SearchBookmark sb = null;

      rs = searchDevice(dst, sb);
      sb = rs.getSearchBookmark();

      while (sb != null)
      {
          // print out the data in the record search result.
          sb = printRecordSearchResults(rs);

          // call the search routine again
          rs = searchDevice(dst, sb);
      }
 }
 private static RecordSearchResults searchDevice(DeviceSearchType dst,
                SearchBookmark sb) throws Exception  {
      RecordSearchResults rs = null;
      final Batch batch = s_conn.newBatch();
      final int numberOfRecordReturn = 10;

      //calling the search API
      batch.searchDevice(dst, sb, numberOfRecordReturn);

      // Call the RDU.
      BatchStatus batchStatus = batch.post();

      // Check for success.
      CommandStatus commandStatus = null;

      if (0 < batchStatus.getCommandCount())
      {
          commandStatus = batchStatus.getCommandStatus(0);
      }
      //check to see if there is an error
      if (batchStatus.isError()
      || batchStatus.isWarning()
      || commandStatus == null
      || commandStatus.isError())
      {
          System.out.println("report batch error.");
          return null;
      }
```

```
      //batch success without error, retrieve the result
      //this is a list of devices
      rs = (RecordSearchResults)commandStatus.getData();
      return rs;
 }


 private static SearchBookmark printRecordSearchResults(RecordSearchResults rs)
throws Exception  {

      SearchBookmark sb = rs.getSearchBookmark();

      List<RecordData> rdlist = rs.getRecordData();
      Iterator<RecordData> iter = rdlist.iterator();

      while (iter.hasNext())
      {
          RecordData rdObj = iter.next();
          Key keyObj = rdObj.getPrimaryKey();

           System.out.println("DeviceOID: " + ((DeviceID)keyObj).getDeviceId());

          //this is for secondary keys.
          List<Key> deviceList = rdObj.getSecondaryKeys();

          if (deviceList != null && !deviceList.isEmpty())
          {
              for (int i=0; i<deviceList.size(); i++)
              {
                  Key key = deviceList.get(i);
                  System.out.println("DeviceID : " + key.toString());
              }
          }
      }
      return sb;

 }
```

ベンダー プレフィックスまたはサービス クラスを使用したデバイスの検索

サービス プロバイダーは管理者に対して、特定のベンダー プレフィックスまたはサービス クラスに一致するすべてのデバイスを検索することを許可します。

目的

サービス プロバイダーの管理アプリケーションは、要求されたベンダー プレフィックスまたはサービス クラスに一致するデバイスのリストを返します。

ステップ 1　管理者は、サービス プロバイダーの管理者のユーザ インターフェイスに、目的のベンダー プレフィックスに一致する部分文字列を入力します。

ステップ 2　BAC は、要求されたベンダー プレフィックスまたはサービス クラスに一致するデバイスの MAC アドレスすべてのリストを、組み込みデータベースにクエリーします。この例では、検索クエリーを組み立て、MAC アドレスを使用してデバイスを取得する方法を示します。P.D-40 の「デバイス タイプを使用した検索」も参照してください。

```
DeviceIDPattern pattern = new MACAddressPattern("1,6,22:49:*");

DeviceSearchType dst = DeviceSearchType.getDevices(pattern, ReturnParameters.ALL);

// To set up search for class of service:

DeviceSearchType searchType = DeviceSearchType.getByClassOfService(

new ClassOfServiceName(name), AssociationType

.valueOf(association), ReturnParameters.ALL);
```

ステップ 3　サービス プロバイダーのアプリケーションは、これらのデバイスの詳細を BAC に要求し、デバイス データのページを表示します。デバイスごとに、デバイス タイプ、MAC アドレス、クライアント クラス、およびデバイスのプロビジョニング ステータスが表示されます。1 行にデバイスが 1 つ表示されます。

```
// calling the search procedure

rs = searchDevice(connection, dst, sb);
sb = processRecordSearchResults(rs);

if (rs != null)
{
    while (sb != null)
         {
            // The search returns a search bookmark, which can be used to make
            // the next search call that would return next set of results

            rs = searchDevice(connection, dst, sb);
            sb = processRecordSearchResults(rs);
        }
    }
}
}
```

PacketCable eMTA の事前プロビジョニング

新しい顧客は、サービス プロバイダーに連絡して PacketCable 音声サービスを注文します。顧客は、プロビジョニングされた組み込み型 MTA を受け取ります。

目的

次のワークフローを使用して、組み込み型 MTA を事前プロビジョニングし、モデムの MTA コンポーネントがオンラインになったときに適切なレベルのサービスを受けられるようにします。

（注） 次の使用例では、eMTA から電話をかける際に必要なコール エージェントのプロビジョニングを省略しています。

ステップ 1 サービス　プロバイダーは、課金システムで使用される加入者のユーザ名とパスワードを選択します。

ステップ 2 サービス プロバイダーは、モデム コンポーネントに適切なサービス クラスと DHCP 基準を選択し、そのコンポーネントを BAC に追加します。

```
// Create a new connection

PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");


Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION);

// Let’s provision the modem and the MTA component in the same
// batch. This can be done because the activation mode of this
// batch is NO_ACTIVATION. More than one device can be operated
// on in a batch if the activation mode does not lead to more
// than one device being reset.
// To add a DOCSIS modem:

List<DeviceID> modemDeviceIDList = new ArrayList<DeviceID>();
modemDeviceIDList.add(new MACAddress("1,6,01:02:03:04:05:06"));
batch.add(
    DeviceType.DOCSIS,      // deviceType: DOCSIS
    modemDeviceIDList,      // macAddress: scanned from the label
    null,                   // hostName: not used in this example
    null,                   // domainName: not used in this example
    "0123-45-6789",         // ownerID: here, account number from billing system
    "Silver",               // classOfService
    "provisionedCM",        // DHCP Criteria: Network Registrar uses this to
                            // select a modem lease granting provisioned IP address
    null                    // properties: not used
);
```

ステップ 3　サービス プロバイダーは、MTA コンポーネントに適切なサービス クラスと DHCP 基準を選択し、そのコンポーネントを BAC に追加します。

```
    List<DeviceID> packetcableMTADeviceIDList = new ArrayList<DeviceID>();
    packetcableMTADeviceIDList.add(new MACAddress("1,6,01:02:03:04:05:07"));

    // Continuation of the batch in Step2
    // To add the MTA component:

    batch.add(
        DeviceType.PACKET_CABLE_MTA,  // deviceType: PACKET_CABLE_MTA
        packetcableMTADeviceIDList,   // macAddress: scanned from the label
        null,                         // hostName: not used in this example, will be
        auto                               // generated
        null,                         // domainName: not used in this example, will be
                                      // auto generated. The FqdnKeys.AUTO_FQDN_DOMAIN
                                      // property must be set somewhere in the property
                                      // hierarchy.
        "0123-45-6789",               // ownerID: here, account number from billing
        system
        "Silver",                     // ClassOfService
        "provisionedMTA",             // DHCP Criteria: Network Registrar uses this to
                                      // select an MTA lease granting provisioned IP
                                      // address
        null                          // properties: not used
    );

    BatchStatus batchStatus = null;

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 4　組み込み型 MTA が顧客に出荷されます。

ステップ 5　顧客は組み込み型 MTA をオンラインにして、電話をかけます。

PacketCable eMTA 上での SNMP クローニング

管理者は SNMP Element Manager アクセス権を PacketCable eMTA に付与します。

目的

外部の Element Manager に、PacketCable eMTA へのセキュアな SNMPv3 アクセス権を付与します。

（注）　RW MIB 変数に加えた変更は永続的なものではなく、BAC の eMTA に関する設定では更新されません。eMTA MIB に書き込まれた情報は、次回 MTA を電源オフにするか、またはリセットしたときに失われます。

ステップ 1　プロビジョニング API メソッドである *performOperation(...)* を呼び出します。その際、MTA の MAC アドレスと、MTA 上に作成する新しいユーザのユーザ名を渡します。このユーザ名は、後で Element Manager が SNMP コールを行うときに使用されます。

```
    // Create a new connection
    PACEConnection conn = PACEConnectionFactory.getInstance(
        "localhost", 49187, "admin", "admin123");

    Batch batch = conn.newBatch(

        // No reset

        ActivationMode.NO_ACTIVATION,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION);

    // NO_ACTIVATION is the activation mode because we don't want to
    // reset the device.
    // NO_CONFIRMATION is the confirmation mode because we are
    // not attempting to reset the device.
    // The goal here is to create a new user on the MTA indicated
    // by the MAC address. The other parameter needed here is the new
    // user name, which is passed in the Map.
    // Create a map that contains one element - the name of
    // the new user to be created on the MTA

    HashMap<String, Object> map = new HashMap<String, Object>();
    map.put( SNMPPropertyKeys.CLONING_USERNAME, "newUser" );

    // The first param is the actual device operation to perform.

    batch.performOperation(
        DeviceOperation.ENABLE_SNMPV3_ACCESS,    // deviceOperation :
        ENABLE_SNMPV3_ACCESS
        new MACAddress("1,6,00:00:00:00:00:99"), // macORFqdn : MAC Address of the
        modem
        map                                      // parameters: operation specific
                                                 // parameters
    );

    BatchStatus batchStatus = null;

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 2　プロビジョニング API は、ステップ 1 で渡された新しいユーザに対応するエントリを MTA 上に作成するため、SNMPv3 クローニング操作の実行を試みます。新しいユーザのエントリ行で使用されるキーは、BAC 内で定義された 2 つのパスワードの関数です。顧客はこれらのパスワードを使用できるようになります。また、RDU コマンドは、これらのパスワード（auth および priv パスワード）をキー ローカリゼーション アルゴリズムに渡して、auth キーおよび priv キーを作成します。これらのキーは、新しいユーザとセットで、eMTA のユーザ テーブルに格納されます。

（注） このステップに記載されている auth および priv パスワードを変更するには、*rdu.properties* 設定ファイル内の `SNMPPropertyKeys.CLONING_AUTH_PASSWORD`（*/snmp/cloning/auth/password*）および `SNMPPropertyKeys.CLONING_PRIV_PASSWORD`（*/snmp/cloning/priv/password*）プロパティをそれぞれ設定変更します。

ステップ 3　顧客は、指定されたユーザ名、パスワード、およびキー ローカリゼーション アルゴリズムを使用して SNMPv3 要求を発行し、MTA とのセキュアな通信を可能にします。

PacketCable eMTA の差分プロビジョニング

顧客は、PacketCable eMTA を使用しており、その 1 本目の回線（エンドポイント）をイネーブルにしています。この顧客は、eMTA 上の 2 本目の電話回線（エンドポイント）をイネーブルにし、その回線に電話を接続します。

目的

顧客は eMTA 上の 2 本目の回線（エンドポイント）に電話を接続し、どのサービスも中断することなく正常に電話をかけられるようになる必要があります。

（注） eMTA 上で 2 本目の回線を使用するには、コール エージェントを適切に設定する必要があります。コール エージェントのプロビジョニングについては、この使用例では取り上げません。

ステップ 1　サービス プロバイダーのアプリケーションは、BAC API を起動して、eMTA のサービス クラスを変更します。新しいサービス クラスは、eMTA 上の 2 つのエンドポイントをサポートします。このサービス クラスの変更は、eMTA がリセットされると有効になります。eMTA を中断することは好ましくありません。そのため、次のステップで差分プロビジョニングが行われます。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION);

// NO_ACTIVATION is the activation mode because we don't want to
// reset the device.
// NO_CONFIRMATION is the Confirmation mode because we are not
// disrupting the device.

batch.changeClassOfService(
    new MACAddress("1,6,ff:00:ee:11:dd:22"),  / eMTA's MAC address or FQDN
    "twoLineEnabledCOS"                      // This COS supports two lines.
);
```

```
BatchStatus batchStatus = null;

// post the batch to RDU server

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
}
```

ステップ 2　サービス プロバイダーのアプリケーションは、BAC 差分更新機能を使用して、eMTA に SNMP オブジェクトを設定します。そのため、eMTA を中断することなく、サービスがイネーブルになります。

```
// The goal here is to enable a second phone line, assuming one
// phone line is currently enabled.   We will be adding a new
// row to the pktcNcsEndPntConfigTable.

batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION);

// NO_ACTIVATION is the activation mode because we don’t want to
// reset the device.
// NO_CONFIRMATION is the confirmation mode because we are
// not attempting to reset the device.
// Create a map containing one element - the list of SNMP
// variables to set on the MTA

HashMap<String, Object> map = new HashMap<String, Object>();

// Create an SnmpVarList to hold SNMP varbinds

SnmpVarList list = new SnmpVarList();

// An SnmpVariable represents an oid/value/type triple.
// pktcNcsEndPntConfigTable is indexed by the IfNumber, which in this case we will
// assume is interface number 12 (this is the last number in each of the oids
below).
// The first variable represents the creation of a new row in
// pktcNcsEndPntConfigTable we are setting the RowStatus
// column (column number 26). The value of 4 indicates that
// a new row is to be created in the active state.

SnmpVariable variable = new SnmpVariable( ".1.3.6.1.4.1.4491.2.2.2.1.2.1.1.26.12",
    "4", SnmpType.INTEGER );
list.add( variable );

// The next variable represents the call agent id for this new
// interface, which we'll assume is 'test.com'

variable = new SnmpVariable( ".1.3.6.1.4.1.4491.2.2.2.1.2.1.1.1.12", "test.com",
    SnmpType.STRING );
list.add( variable );

// The final variable represents the call agent port
```

```
    variable = new SnmpVariable( ".1.3.6.1.4.1.4491.2.2.2.1.2.1.1.2.12", "2728",
        SnmpType.INTEGER );
    list.add( variable );

    // Add the SNMP variable list to the Map to use in the API call

    map.put( SNMPPropertyKeys.SNMPVAR_LIST, list );

    // Invoke the BACC API to do incremental update on the eMTA.

    batch.performOperation(
        DeviceOperation.INCREMENTAL_UPDATE,      // device operation
        new MACAddress("1,6,00:00:00:00:00:99"), // MAC Address
        map                                      // Parameters for the operation
    );

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 3　eMTA で 2 本目の電話回線を使用できるようになります。ステップ 1 でサービス クラスが変更されているので、eMTA はリセット後も引き続き同じサービスを受けられます。

動的設定ファイルを使用した DOCSIS モデムの事前プロビジョニング

新しい顧客は、サービス プロバイダーに連絡して、モデムの背後に接続された 2 台の CPE を対象とする高速な *Gold* データ サービス付きの DOCSIS モデムを注文します。

目的

次のワークフローを使用して、DOCSIS テンプレートを使用するサービス クラスで DOCSIS モデムを事前プロビジョニングします。テンプレートから生成される動的設定ファイルは、モデムがオンラインになるときに使用されます。

ステップ 1　サービス　プロバイダーは、課金システムで使用される加入者のユーザ名とパスワードを選択します。

ステップ 2　サービス プロバイダーは、Gold サービス クラスと適切な DHCP 基準を選択し、ケーブル モデムを BAC に追加します。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION);


Map<String, Object> properties = new HashMap<String, Object>();

// Set the property PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED to enable
// promiscuous mode on modem

properties.put(PolicyKeys.COMPUTER_PROMISCUOUS_MODE_ENABLED, Boolean.TRUE);

// enable promiscuous mode by changing the technology default

batch.changeDefaults(DeviceType.DOCSIS,properties, null);

BatchStatus batchStatus = null;

// post the batch to RDU server

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
}
// No CPE DHCP Criteria is specified.
// The CPE behind the modem will use the default provisioned
// promiscuous CPE DHCP criteria specified in the system defaults.
// This custom property corresponds to a macro variable in the
// DOCSIS template for “gold” class of service indicating the
// maximum number of CPE allowed behind this modem. We set it
// to two sets of CPE from this customer.

properties = new HashMap<String, Object>();
properties.put("docsis-max-cpes", "2");

batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// To add a DOCSIS modem:

List<DeviceID> deviceIDList = new ArrayList<DeviceID>();
```

```
    deviceIDList.add(new MACAddress("1,6,01:02:03:04:05:06"));
    batch.add(
        DeviceType.DOCSIS,      // deviceType: DOCSIS
        deviceIDList,           // macAddress: scanned from the label
        null,                   // hostName: not used in this example
        null,                   // domainName: not used in this example
        "0123-45-6789",         // ownerID: here, account number from billing system
        "gold",                 // classOfService:
        "provisionedCM",        // DHCP Criteria: Network Registrar uses this to
                                // select a modem lease granting provisioned IP address
        properties              // properties:
    );

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 3　ケーブル モデムが顧客に出荷されます。

ステップ 4　顧客はケーブル モデムをオンラインにし、モデムの背後にコンピュータを接続します。

オプティミスティック ロッキング

サービス プロバイダー アプリケーションのインスタンスは、同じアプリケーションの別のインスタンスによって行われた変更を上書きしないようにします。

目的

次のワークフローを使用して、BAC API が提供するオプティミスティック ロッキング機能を実行します。

（注）　オブジェクトのロッキングは、マルチユーザのシステムで変更の整合性を維持するために実行されます。その結果、ユーザの変更は別のユーザによって不用意に上書きされなくなります。オプティミスティック ロッキングを使用してプログラムを記述する場合、コミットしようとするオブジェクトが 1 つでも処理の開始後に別のユーザによって変更されていたときは、コミットが失敗する可能性があります。

ステップ 1　サービス担当者は、サービス プロバイダーのユーザ インターフェイスで検索オプションを選択し、ケーブル モデムの MAC アドレスを入力します。

ステップ 2　BAC は、組み込みデータベースにクエリーし、デバイスの詳細を取得します。次に、その情報が MSO ユーザ インターフェイスに表示されます。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);


final DeviceID modemMACAddress = DeviceID.getInstance("1,6,00:11:22:33:44:55",
        KeyType.MAC_ADDRESS);
List<DeviceDetailsOption> options = new ArrayList<DeviceDetailsOption>();

options.add(DeviceDetailsOption.INCLUDE_LEASE_INFO);

// MSO admin UI calls the provisioning API to query the details
// for the requested device. Query may be performed based on MAC
// address or IP address, depending on what is known about the
// device.

batch.getDetails(modemMACAddress, options);

// post the batch to RDU server

BatchStatus batchStatus = null;

try
{
    batchStatus = batch.post();
}
catch(ProvisioningException e)
{
    e.printStackTrace();
}
```

ステップ 3　サービス担当者は、ユーザ インターフェイスを使用して、モデムのサービス クラスと DHCP 基準の変更を試みます。その結果、BAC API が起動します。

```
    Map<String, Object> deviceDetails = new TreeMap((Map<String,
        Object>)batchStatus.getCommandStatus(0).getData());

    //  extract device detail data from the map

    String deviceType = (String)deviceDetails.get(DeviceDetailsKeys.DEVICE_TYPE);
    String macAddress = (String)deviceDetails.get(DeviceDetailsKeys.MAC_ADDRESS);
    String relayAgentID = (String)deviceDetails.get(DeviceDetailsKeys.RELAY_AGENT_MAC);
    Boolean isProvisioned =
    (Boolean)deviceDetails.get(DeviceDetailsKeys.IS_PROVISIONED);

    //  Let's save the OID_REVISION_NUMBER property so that we can set it in
    //  step 3.

    String oidRevisionNumber =
        (String)deviceDetails.get(GenericObjectKeys.OID_REVISION_NUMBER);

    // We need a reference to Batch instance so that ensureConsistency()
    // method can be invoked on it.

    batch = conn.newBatch() ;
    List<String> oidList = new ArrayList<String>();

    // Add the oid-rev number saved from step 2 to the list

    oidList.add(oidRevisionNumber);

    // Sends a list of OID revision numbers to validate before processing the
    // batch. This ensures that the objects specified have not been modified
    // since they were last retrieved.

    batch.ensureConsistency(oidList);
    batch.changeClassOfService (
        new MACAddress("1,6,00:11:22:33:44:55"),  // macORFqdn: unique identifier for
        the                                           // device.
        "gold"                                    // newCOSName : Class of service
        name.
    );

    batch.changeDHCPCriteria (
        new MACAddress("1,6,00:11:22:33:44:55"),  // macORFqdn: unique identifier for
        the                                           // device.
        "specialDHCPCriteria"                     // newDHCPCriteria : New DHCP
        Criteria.
    );

    // This batch fails with BatchStatusCodes.BATCH_NOT_CONSISTENT,
    // in case if the device is updated by another client in the meantime.
    // If a conflict occurs, then the service provider client
    // is responsible for resolving the conflict by querying the database
    // again and then applying changes appropriately.
    }
}
```

ステップ 4　これで、ユーザは適切な DHCP 基準の Gold サービス クラスを受けられるようになります。

加入者の帯域幅の一時的なスロットリング

MSO は、加入者に許可するダウンロードのデータ量を 1 か月あたり 10 MB に制限するサービスを提供します。その限度に達すると、加入者のダウンストリーム帯域幅が 10 MB から 56 KB に制限されます。月が替わると 10 MB に戻ります。

（注） ピアツーピア ユーザや Web サイトを運営するユーザはアップロード帯域幅の使用量が非常に高くなる傾向があるため、アップストリーム帯域幅の変更も考慮することが必要になる場合があります。

目的

次のワークフローを使用して、加入者の帯域幅をその契約条件に従って増減させます。

ステップ 1　MSO は *NetFlow* などのレート追跡システムを使用します。このシステムは、MAC アドレスに基づいて各顧客の使用率を追跡します。最初に、顧客は、1 MB のダウンストリームを持つ *Gold* サービス クラスのレベルでプロビジョニングされます。

ステップ 2　レート追跡ソフトウェアは、加入者が 10 MB の限度に達したことを特定すると、OSS に通知します。OSS は BAC API を呼び出して、加入者のサービス クラスを *Gold* から *Gold-throttled* に変更します。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);


// AUTOMATIC is the activation mode because we are
// attempting to reset the modem so that it
// receives low bandwidth service.
// NO_CONFIRMATION is the confirmation mode
// because we do not want the batch to fail if we cannot
// reset the modem. If the modem is off, then it will
// be disabled when it is turned back on.
// Let’s change the COS of the device so that it restricts
// bandwidth usage of the modem.

batch.changeClassOfService(
    new MACAddress("1,6,00:11:22:33:44:55"),  // macAddress: unique identifier for
                                              // this modem
    "Gold-throttled"                          // newClassOfService: restricts
                                              // bandwidth usage to 56k
);

BatchStatus batchStatus = null;
```

```
    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 3　請求対象期間の終了時に、OSS は BAC API を呼び出して、加入者のサービス クラスを *Gold* に戻します。

CableHome WAN-MAN の事前プロビジョニング

新しい顧客は、サービス プロバイダーに連絡してホーム ネットワーキング サービスを注文します。顧客は、プロビジョニングされた CableHome デバイスを受け取ります。

目的

次のワークフローを使用して、CableHome デバイスを事前プロビジョニングし、ケーブル モデムとその WAN-MAN コンポーネントがオンラインになったときに適切なレベルのサービスを受けられるようにします。

ステップ 1　サービス　プロバイダーは、課金システムで使用される加入者のユーザ名とパスワードを選択します。

ステップ 2　サービス プロバイダーは、モデム コンポーネントに適切なサービス クラスと DHCP 基準を選択し、そのコンポーネントを BAC に追加します。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation
    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// Let's provision the modem and the WAN-Man component in the same
// batch.
// To add a DOCSIS modem:

List<DeviceID> docisDeviceIDList = new ArrayList<DeviceID>();
docisDeviceIDList.add(new MACAddress("1,6,01:02:03:04:05:06"));
batch.add(
    DeviceType.DOCSIS,      // deviceType: DOCSIS
    docisDeviceIDList,      // macAddress: scanned from the label
    null,                   // hostName: not used in this example
    null,                   // domainName: not used in this example
    "0123-45-6789",         // ownerID: here, account number from billing system
    "Silver",               // classOfService
    "provisionedCM",        // DHCP Criteria: Network Registrar uses this to
                            // select a modem lease granting provisioned IP address
    null                    // properties: not used
);
```

ステップ 3　サービス プロバイダーは、WAN-MAN コンポーネントに適切なサービス クラスと DHCP 基準を選択し、そのコンポーネントを BAC に追加します。

```
List<DeviceID> wanManDeviceIDList = new ArrayList<DeviceID>();
wanManDeviceIDList.add(new MACAddress("1,6,01:02:03:04:05:07"));
batch.add(
    DeviceType.CABLEHOME_WAN_MAN,   // deviceType: CABLEHOME_WAN_MAN
    wanManDeviceIDList,             // macAddress: scanned from the label
    null,                           // hostName: not used in this example
    null,                           // domainName: not used in this example
    "0123-45-6789",                 // ownerID: here, account number from billing
                                    // system
    "silverWanMan",                 // classOfService
    "provisionedWanMan",            // DHCP Criteria: Network Registrar uses this
    to
                                   // select a modem lease granting provisioned IP
                                    // address
    null                            // properties: not used
);
}
```

ステップ 4　CableHome デバイスが顧客に出荷されます。

ステップ 5　顧客は CableHome デバイスをオンラインにします。

ファイアウォール設定を持つ CableHome

顧客はサービス プロバイダーに連絡して、ファイアウォール機能がイネーブルになったホームネットワーキング サービスを注文します。顧客は、プロビジョニングされた CableHome デバイスを受け取ります。

目的

次のワークフローを使用して、CableHome デバイスを事前プロビジョニングし、ケーブル モデムとその WAN-MAN コンポーネントがオンラインになったときに適切なレベルのサービスを受けられるようにします。

ステップ 1　サービス　プロバイダーは、課金システムで使用される加入者のユーザ名とパスワードを選択します。

ステップ 2　サービス プロバイダーは、ケーブル モデム コンポーネントに適切なサービス クラスと DHCP 基準を選択し、そのコンポーネントを BAC に追加します。

```
PACEConnection conn = PACEConnectionFactory.getInstance(
    "localhost", 49187, "admin", "admin123");

Batch batch = conn.newBatch(

    // No reset

    ActivationMode.NO_ACTIVATION,

    // No need to confirm activation

    ConfirmationMode.NO_CONFIRMATION,

    // No publishing to external database

    PublishingMode.NO_PUBLISHING);

// Let's provision the modem and the WAN-Man component in the same
// batch.
// To add a DOCSIS modem:

List<DeviceID> docisDeviceIDList = new ArrayList<DeviceID>();
docisDeviceIDList.add(new MACAddress("1,6,01:02:03:04:05:06"));
batch.add(
    DeviceType.DOCSIS,      // deviceType: DOCSIS
    docisDeviceIDList,      // macAddress: scanned from the label
    null,                   // hostName: not used in this example
    null,                   // domainName: not used in this example
    "0123-45-6789",         // ownerID: here, account number from billing system
    "Silver",               // classOfService
    "provisionedCM",        // DHCP Criteria: Network Registrar uses this to
                            // select a modem lease granting provisioned IP address
    null                    // properties: not used
);
```

ステップ 3　サービス プロバイダーは、WAN-MAN コンポーネントに適切なサービス クラスと DHCP 基準を選択し、そのコンポーネントを BAC に追加します。

```
    //      Continuation of the batch in Step 2
    // To add the WAN-Man component:
    // Create a Map to contain WanMan's properties

    Map<String, Object> properties = new HashMap<String, Object>();

    // The fire wall configuration for the Wan Man component is specified
    // using the CableHomeKeys.CABLEHOME_WAN_MAN_FIREWALL_FILE property.
    // This use case assumes that the firewall configuration file named
    // "firewall_file.cfg" is already present in the RDU database and the
    // firewall configuration is enabled in the Wan Man configuration file
    // specified with the corresponding class of service.

    properties.put(CableHomeKeys.CABLEHOME_WAN_MAN_FIREWALL_FILE, "firewall_file.cfg");

    List<DeviceID> wanManDeviceIDList = new ArrayList<DeviceID>();
    wanManDeviceIDList.add(new MACAddress("1,6,01:02:03:04:05:07"));
    batch.add(
        DeviceType.CABLEHOME_WAN_MAN,    // deviceType: CABLEHOME_WAN_MAN
        wanManDeviceIDList,              // macAddress: scanned from the label
        null,                          // hostName: not used in this example
        null,                          // domainName: not used in this example
        "0123-45-6789",                // ownerID: here, account number from billing
        system
        "silverWanMan",                // classOfService
        "provisionedWanMan",           // DHCP Criteria: Network Registrar uses this to
                                       // select a modem lease granting provisioned IP
                                       // address
        null                           // properties: not used
    );

    BatchStatus batchStatus = null;

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 4　CableHome デバイスが顧客に出荷されます。

ステップ 5　顧客は CableHome デバイスをオンラインにします。次に、ケーブル モデムと WAN-MAN コンポーネントは、プロビジョニングされた IP アドレスと正しい設定ファイルを取得します。

CableHome WAN-MAN のデバイス機能の取得

サービス プロバイダーは管理者に対して、CableHome WAN-MAN デバイスの機能の情報を表示することを許可します。

目的

サービス プロバイダーの管理アプリケーションは、特定の CableHome WAN-MAN コンポーネントに関する既知の詳細をすべて表示します。この詳細には、MAC アドレス、リース情報、プロビジョニング ステータス、およびデバイス機能の情報などがあります。

ステップ 1　管理者は、サービス プロバイダーのユーザ インターフェイスに、クエリーする WAN-MAN の MAC アドレスを入力します。

ステップ 2　BAC は、入力された MAC アドレスを使用して特定したデバイスの詳細を、組み込みデータベースにクエリーします。

```
    PACEConnection conn = PACEConnectionFactory.getInstance(
        "localhost", 49187, "admin", "admin123");


    Batch batch = conn.newBatch(

        // No reset

        ActivationMode.NO_ACTIVATION,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION,

        // No publishing to external database

        PublishingMode.NO_PUBLISHING);


    final DeviceID modemMACAddress = DeviceID.getInstance("1,6,00:11:22:33:44:55",
            KeyType.MAC_ADDRESS);

    List<DeviceDetailsOption> options = new ArrayList<DeviceDetailsOption>();
    options.add(DeviceDetailsOption.INCLUDE_LEASE_INFO);

    // MSO admin UI calls the provisioning API to query the details
    // for the requested device. Query may be performed based on MAC
    // address or IP address, depending on what is known about the
    // device.

    batch.getDetails(modemMACAddress, options);

    // post the batch to RDU server

    BatchStatus batchStatus = null;
    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 3　サービス プロバイダーのアプリケーションは、デバイス データの詳細に関するページを表示します。このページには、要求されたデバイスに関する既知の情報をすべて表示できます。デバイスがサービス プロバイダーのネットワークに接続された場合、このデータにはリース情報（IP アドレスやリレー エージェントの ID など）が含まれます。このデータは、デバイスがプロビジョニングされているかどうかを示します。プロビジョニングされている場合、データにはデバイス タイプとデバイス機能の情報も含まれます。

```
    Map<String, Object> deviceDetails = new TreeMap((Map<String,
        Object>)batchStatus.getCommandStatus(0).getData());

    //  extract device detail data from the map

    String deviceType = (String)deviceDetails.get(DeviceDetailsKeys.DEVICE_TYPE);
    String macAddress = (String)deviceDetails.get(DeviceDetailsKeys.MAC_ADDRESS);
    String relayAgentID = (String)deviceDetails.get(DeviceDetailsKeys.RELAY_AGENT_MAC);
    Boolean isProvisioned =
    (Boolean)deviceDetails.get(DeviceDetailsKeys.IS_PROVISIONED);

    String deviceID = (String) deviceDetails.get(CNRNames.DEVICE_ID.toString());
    String serNum = (String)
    deviceDetails.get(CNRNames.DEVICE_SERIAL_NUMBER.toString());
    String hwVer = (String)
        deviceDetails.get(CNRNames.HARDWARE_VERSION_NUMBER.toString());
    String swVer = (String)
        deviceDetails.get(CNRNames.SOFTWARE_VERSION_NUMBER.toString());
    String brVer = (String) deviceDetails.get(CNRNames.BOOT_ROM_VERSION.toString());
    String vendorOui = (String) deviceDetails.get(CNRNames.VENDOR_OUI.toString());
    String modelNum = (String) deviceDetails.get(CNRNames.MODEL_NUMBER.toString());
    String vendorNum = (String) deviceDetails.get(CNRNames.VENDOR_NAME.toString());

    //  The admin UI now formats and prints the detail data to a view page
    }
}
```

CableHome WAN-MAN のセルフプロビジョニング

加入者は、戸建住宅内にブラウザ アプリケーションを持つコンピュータを設置し、組み込み CableHome デバイスを購入しました。

目的

次のワークフローを使用して、プロビジョニングされていない新しい組み込み CableHome デバイスをオンラインにし、適切なレベルのサービスを受けられるようにします。さらに、組み込み CableHome デバイスに接続されたコンピュータから加入者がインターネットにアクセスできるようにします。

ステップ 1　加入者は組み込み CableHome デバイスを購入し、住宅内に設置します。

ステップ 2　加入者は、組み込み CableHome デバイスの電源をオンにします。BAC は、制限付きのアクセス権を組み込みケーブル モデムに付与します。つまり、アクセス権は 2 台の CPE（CableHome WAN-MAN とコンピュータ）に制限されます。

（注） この使用例は、プロビジョニングされていない DOCSIS モデムの背後に 2 台の CPE を接続できることを前提としています。特に設定しない限り、BAC では、プロビジョニングされていない DOCSIS モデムの背後に接続できるデバイスは 1 台のみです。この動作を変更するには、2 台の CPE をサポートする適切なサービス クラスを定義してから、そのサービス クラスを DOCSIS デバイスのデフォルト サービス クラスとして使用します。

ステップ 3 BAC は CableHome WAN-MAN を設定します。この設定には、IP 接続や、デフォルトの CableHome ブート ファイルのダウンロードなどがあります。デフォルトの CableHome ブート ファイルは、CableHome デバイスをパススルー モードに設定します。CableHome デバイスはまだプロビジョニングされていません。

ステップ 4 加入者は、コンピュータを CableHome デバイスに接続します。コンピュータは、プロビジョニングされていない（制限された）IP アドレスを取得します。加入者はコンピュータ上でブラウザ アプリケーションを起動します。スプーフィング DNS サーバにより、ブラウザがサービス プロバイダーの登録サーバ（OSS ユーザ インターフェイスやメディエータなど）にアクセスします。

ステップ 5 加入者はサービス プロバイダーのユーザ インターフェイスを使用して、ケーブル モデムの登録に必要な手順（サービス クラスの選択など）を終了します。また、加入者は CableHome のサービス クラスも選択します。

ステップ 6 サービス プロバイダーのユーザ インターフェイスは、加入者の情報（ケーブル モデムと CableHome に対して選択されたサービス クラスやコンピュータの IP アドレスなど）を BAC に渡します。次に、加入者が BAC に登録されます。

ステップ 7 ユーザ インターフェイスは、加入者にコンピュータをリブートするように求めます。

ステップ 8 プロビジョニング クライアントは *performOperation(…)* を呼び出してモデムをリブートし、プロビジョニングされたアクセス権をモデムに付与します。

```
    // create a new batch

    batch = conn.newBatch(

        // No reset

        ActivationMode.AUTOMATIC,

        // No need to confirm activation

        ConfirmationMode.NO_CONFIRMATION);


    // register performOperation command to the batch

    batch.performOperation(DeviceOperation.RESET, modemMACAddressObject, null);

    // post the batch to RDU server

    try
    {
        batchStatus = batch.post();
    }
    catch(ProvisioningException e)
    {
        e.printStackTrace();
    }
}
```

ステップ 9　コンピュータをリブートすると、コンピュータは CableHome デバイスの DHCP サーバから新しい IP アドレスを受信します。これで、ケーブル モデムと CableHome デバイスはいずれもプロビジョニングされます。この結果、加入者は CableHome デバイスのイーサネット ポートに複数のコンピュータを接続し、インターネットにアクセスすることができます。

（注）　WAN-MAN コンポーネントに提供された設定ファイルによってボックスの WAN-Data コンポーネントがイネーブルにされる場合、コンポーネントは無差別モードでプロビジョニングされます。無差別モードは、DeviceType.CABLEHOME_WAN_DATA デバイス タイプのテクノロジー デフォルト レベルでイネーブルになることが前提となっています。

■ 使用例

APPENDIX **E**

Broadband Access Center のプロビジョニングに関する FAQ

この付録では、BAC のプロビジョニングに関する FAQ への回答を示します。

BAC の設定

この項では、一般的な BAC の設定に関する FAQ を取り上げます。

- Registrar 拡張をイネーブルまたはディセーブルにするにはどうすればよいですか？
- Network Registrar 拡張のトレースをイネーブルにするにはどうすればよいですか？
- DPE サーバの登録が失敗するのはなぜですか？

Registrar 拡張をイネーブルまたはディセーブルにするにはどうすればよいですか？

この項で説明する手順では、次の事項を前提としています。

- BAC コンポーネントが */opt/CSCObac* にインストールされている。
- Cisco Network Registrar が */opt/nwreg2* にインストールされている。

Network Registrar 拡張ポイントを手動でインストールするには、次の手順を実行します。

ステップ 1　Network Registrar サーバに、*root* アクセス権でログインします。

ステップ 2　*libbprextensions.so* ディレクトリを *NR_HOME/local/extensions/dhcp/dex/* ディレクトリにコピーします。

ステップ 3　*cnr_ep.properties* ファイルを *BPR_HOME/cnr_ep/conf* ディレクトリにコピーします。

ステップ 4　Network Registrar コマンドライン ツール（**nrcmd**）から、次のように拡張を設定します。

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

Network Registrar 拡張ポイントを手動でディセーブルにするには、次の手順を実行します。

ステップ 1　Network Registrar サーバに、*root* アクセス権でログインします。

ステップ 2　次のように入力します。

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

ステップ 3　*NR_HOME/local/extensions/dhcp/dex/* ディレクトリにある *libbprextensions.so* ファイルを削除します。

Network Registrar 拡張のトレースをイネーブルにするにはどうすればよいですか？

Network Registrar 拡張ポイントのトレースをイネーブルにするには、次の手順を実行します。

ステップ 1　Network Registrar の Web UI にログインします。デフォルトのログインとパスワードは **admin** と **changeme** です。

ステップ 2　メニューから、**DHCP > DHCP Server** ページをクリックします。

Manage DHCP Server ページが表示されます。

ステップ 3　DHCP Server リンクをクリックします。

Edit DHCP Server ページが表示されます。

ステップ 4　Extensions カテゴリを展開し、**extension-trace-level** の値を 3 または 4 に設定します。

ステップ 5　着信パケットと発信パケットを表示するには、Logging カテゴリを展開し、**incoming-packet-detail** チェックボックスと **outgoing-packet-detail** チェックボックスをオンにします。

ステップ 6　**Modify Server** をクリックします。

ステップ 7　DHCP サーバをリロードします。

DPE サーバの登録が失敗するのはなぜですか？

DPE サーバの登録が失敗する原因として、DPE がプロビジョニング グループの要件を満たしていないことが考えられます。

DPE のログ ファイルに、次の内容を示すエラー メッセージがあるかどうかを確認します。

- 他の設定もイネーブルにする必要がある。たとえば DPE 上で TFTP サーバをイネーブルにする必要があるなど。
- このリリースの BAC だけで使用可能な機能をイネーブルにするためにサーバをアップグレードする必要がある。

IPv6 の設定

この項では、BAC 設定時の IPv6 に関する FAQ を取り上げます。

- DPE の IPv6 プロビジョニングをイネーブルにするにはどうすればよいですか？
- プロビジョニング用に IPv4 インターフェイスを設定するにはどうすればよいですか？
- DPE は IPv6 プロビジョニング用に設定されていますが、BAC は IPv6 DOCSIS 3.0 デバイスをプロビジョニングしません。なぜですか？
- MAC アドレスを使用してすべてのデバイスを検索すると、一部の IPv6 デバイスが表示されません。なぜですか？
- インターフェイス上で IPv6 をイネーブルにするにはどうすればよいですか？
- ループバック インターフェイス上に IPv6 を設定するにはどうすればよいですか？
- Solaris 10 でステートフル DHCPv6 クライアントをディセーブルにするにはどうすればよいですか？
- インターフェイスに固定 IP アドレスを割り当てるにはどうすればよいですか？

DPE の IPv6 プロビジョニングをイネーブルにするにはどうすればよいですか？

DPE の IPv6 プロビジョニングをイネーブルにするには、DPE コマンドラインから次の手順を実行します。

ステップ 1　IPv6 プロビジョニングをイネーブルにするには、次のコマンドを使用して 2 つのインターフェイスを設定する必要があります。

a. DPE が Network Registrar 拡張と通信するときに、IP アドレスで識別される指定のインターフェイスを使用するように DPE を設定するには、次のように入力します。

interface ip *ip_address* **pg-communication**

ip_address：特定の DPE インターフェイスの IPv4 アドレスを示します。

b. IP　アドレスで識別される指定のインターフェイスがプロビジョニング要求を処理するように設定するには、次のように入力します。

interface ip *ip_address* **provisioning**

ip_address：インターフェイスのアドレスを IPv6 形式で指定します。

ステップ 2　次のコマンドを使用して、それぞれのサービスをイネーブルにします。

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

ステップ 3　**dpe reload** コマンドを使用して DPE をリロードします。

プロビジョニング用に IPv4 インターフェイスを設定するにはどうすればよいですか？

プロビジョニング用に IPv4 インターフェイスを設定するには、次のコマンドを使用してそのインターフェイスの Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を設定する必要があります。

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

- *ip_address*：インターフェイスのアドレスを IPv4 形式で指定します。
- *fqdn*：指定されたインターフェイスで設定される FQDN を示します。

DPE は IPv6 プロビジョニング用に設定されていますが、BAC は IPv6 DOCSIS 3.0 デバイスをプロビジョニングしません。なぜですか？

DPE が属するプロビジョニング グループで DOCSIS 3.0 をイネーブルにする必要があります。

BAC 管理者のユーザ インターフェイスから、次の手順を実行します。

ステップ 1　**Servers** > **Provisioning Group** をクリックします。

Provisioning Group Details ページが表示されます。

ステップ 2　特定の DPE に対応する Provisioning Groups リンクをクリックします。

ステップ 3　Capabilities Management 領域で、IPv6 - DOCSIS 3.0 に対応する **Enabled** オプション ボタンをオンにします。

ステップ 4　**Submit** をクリックします。

MAC アドレスを使用してすべてのデバイスを検索すると、一部の IPv6 デバイスが表示されません。なぜですか？

デバイスの MAC アドレス オプションを使用してすべてのデバイスを検索すると、一部の IPv6 デバイスは表示されません。これは、Vista IPv6 コンピュータなどのデバイスが、自分の MAC アドレスを送信要求メッセージで報告しないためです。その結果、このようなデバイスは DUID でしか認識できません。

デバイスが MAC アドレスを CableLabs Device ID オプションで報告する場合、そのデバイスは DUID または MAC アドレスで検索できます。

インターフェイス上で IPv6 をイネーブルにするにはどうすればよいですか？

インターフェイス上で IPv6 をイネーブルにするには、次のコマンドを実行します。

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

intf は、IPv6 をイネーブルにするインターフェイスを示します。

ループバック インターフェイス上に IPv6 を設定するにはどうすればよいですか？

ループバック インターフェイス上に IPv6 を設定する前に、次のコマンドを使用してループバック インターフェイスが稼働していることを確認します。

```
# ifconfig -a
```

ループバック インターフェイスが稼働していない場合、*root* としてログインして次のコマンドを実行します。

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

Solaris 10 でステートフル DHCPv6 クライアントをディセーブルにするにはどうすればよいですか？

Solaris 10 でステートフル DHCPv6 クライアントをディセーブルにするには、次のコマンドを使用して *ndpd.conf* ファイルを変更する必要があります。

```
# cat > /etc/inet/ndpd.conf <<EOF
ifdefault StatefulAddrConf off
EOF
```

インターフェイスに固定 IP アドレスを割り当てるにはどうすればよいですか？

固定 IP アドレスの割り当ては必須ではなりませんが、割り当てるには次のコマンドを実行します。

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

CMTS の設定

この項では、Cable Modem Termination System（CMTS; ケーブル モデム ターミネーション システム）の設定に関する FAQ を取り上げます。

- 両方のケーブル ラインカードがケーブル バンドル 1 を使用していることを確認するにはどうすればよいですか？
- 使用できる IPv6 ケーブル ヘルパー アドレスはありますか？
- IPv4 のプライマリとセカンダリの IPv4 サブネットのように、複数の IPv6 サブネットを設定するにはどうすればよいですか？
- CMTS で IPv6 モデムのリストを表示するにはどうすればよいですか？
- IPv6 シングル スタックだけを受け入れるように CMTS インターフェイスを設定するにはどうすればよいですか？
- モデムの状態 init(x) にはどのような意味があるのですか？

両方のケーブル ラインカードがケーブル バンドル 1 を使用していることを確認するにはどうすればよいですか？

ケーブル インターフェイスごとに次の設定を追加する必要があります。

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

使用できる IPv6 ケーブル ヘルパー アドレスはありますか？

あります。次に示すバンドルの設定が IPv4 のヘルパー アドレスに相当します。

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

IPv4 のプライマリとセカンダリの IPv4 サブネットのように、複数の IPv6 サブネットを設定するにはどうすればよいですか？

IPv6 のバンドルに複数のプレフィックスを割り当てることができますが、IPv6 のサブネットにはプライマリ タイプやセカンダリ タイプはありません。

CMTS で IPv6 モデムのリストを表示するにはどうすればよいですか？

IPv6 モデムのリストを表示するには、次のコマンドを使用します。

```
show cable modem ipv6
```

IPv6 シングル スタックだけを受け入れるように CMTS インターフェイスを設定するにはどうすればよいですか？

ケーブル モデム ターミネーション システム（CMTS）のインターフェイスに次のオプションを追加する必要があります。

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

モデムの状態 init(x) にはどのような意味があるのですか？

show cable modems（scm）コマンドは、接続されているケーブル モデムとそれぞれの状態を表示します。

表 E-1 に、IPv4 と IPv6 のさまざまなモデムの状態を示します。

表 E-1　ケーブル モデムの状態

状態	説明
IPv4	
init(d)	DHCP 検出
init(io)	DHCP オファー
init(dr)	DHCP 要求
init(i)	DHCP 確認
init(o)	TFTP 要求
Init(t)	ToD 要求
online	オンライン
IPv6	
init6(s)	送信要求
init6(a)	アドバタイズ
init6(r)	要求
init6(i)	応答
init6(o)	IPv6 TFTP 要求
init6(t)	IPv6 ToD 要求
online	オンライン

GLOSSARY

A

API
アプリケーション プログラミング インターフェイス（Application programming interface）。サービスへのインターフェイスを定義する関数呼び出し規定の仕様です。

B

BAC
ブロードバンド モデムの構成と管理を行うデータオーバーケーブル サービス プロバイダーの統合ソリューションで、加入者の自動登録とアクティベーションを可能にし、管理します。BAC は、大量のデバイスをサポートできるスケーラブルな製品です。

Broadband Access Center
「BAC」を参照。

Broadband Access Center for Cable
「BAC」を参照。

C

CableHome
CATV 事業者が高品質な付加価値サービスをホーム ローカルエリア ネットワークに展開できるようにするための、標準化されたインフラストラクチャを開発する CableLabs イニシアティブ。

chaddr
DHCP クライアント ハードウェア（MAC）アドレス。クライアントとサーバの間で RFC 2131 パケットを使用して送信されます。

CMTS
ケーブル モデム ターミネーション システム（cable modem termination system）。デジタル信号をケーブル ネットワーク上のケーブル モデムと交換するコンポーネントです。通常はケーブル ヘッドエンドに接続されているルータかブリッジのいずれかです。通常、ケーブル プロバイダーのローカル オフィスにあります。

CMTS 共有秘密情報
「共有秘密情報」を参照。

CPE
宅内装置（customer premises equipment）。電話、コンピュータ、モデムなど、顧客側で用意され、インストールされる着信側機器です。

D

DHCP
Dynamic Host Configuration Protocol。Internet Engineering Task Force（IETF; インターネット技術特別調査委員会）が設計し、TCP/IP を使用する際に必要な設定を減らしました。DHCP は IP アドレスをホストに割り当てます。また、接続されているインターネット ネットワークで情報を操作および交換するのにホストが必要とするすべてのパラメータも提供します。

DNS ドメイン ネーム システム。増加するインターネット ユーザに対応します。DNS は、www.cisco.com などの名前を 192.168.40.0 などの Internet Protocol（IP; インターネット プロトコル）アドレスに変換し、これによってコンピュータが互いに通信できるようになります。

DOCSIS データオーバーケーブル サービス インターフェイス仕様（Data Over Cable Service Interface Specification）。ケーブル テレビ システム ネットワーク上での高速データ配信に関わるケーブル モデムの機能性を定義します。

DOCSIS 共有秘密情報 BAC 配備内の DOCSIS デバイス間の通信用の共有秘密情報。

DPE Device Provisioning Engine。DPE はデバイス情報をキャッシュします。これらの分散型サーバは、RDU と自動的に同期して最新の設定を取得し、BAC のスケーラビリティを実現します。

DSTB デジタル セットトップ ボックス（Digital set-top box）。テレビをインターネットのユーザ インターフェイスにして、デジタル テレビ信号の受信とデコードを行うデバイス。

DUID DHCP Unique Identifier。DHCPv6 でのプライマリ デバイス識別子。

E

eMTA 組み込み型 MTA。MTA とケーブル モデムの両方を含むシングル ノード。

eSAFE embedded Service Application Functional Entity。IPv6 の組み込み型ケーブルおよび IPv4 eMTA から構成される混在 IP モードのデバイス。

F

FQDN Fully qualified domain name。FQDN は、ホスト名以外も含む、システムの完全名です。たとえば、cisco がホスト名で、www.cisco.com が FQDN です。

G

giaddr DHCP ゲートウェイ（リレー エージェント）IP アドレス。クライアントとサーバの間で RFC 2131 パケットを使用して送信されます。

I

Internet Protocol（IP、IPv4） TCP/IP プロトコル スイートのネットワーク層。Internet Protocol（バージョン 4）は、コネクションレス型、ベストエフォート型のパケット スイッチング プロトコルです。RFC 791 で規定されています。

IPv6 IP バージョン 6。現バージョン（バージョン 4）の IP の後続バージョン。IPv6 では、パケット ヘッダー内のフロー ID がサポートされます。この ID を使用して、フローを識別できます。以前は、IPng（next generation）と呼ばれていました。

IP アドレス IP アドレスは、32 ビットの数値で、インターネットにパケットで送信される情報の送信者または受信者を識別します。

K

KDC
限定された Kerberos 機能を実装する、Key Distribution Center（KDC; 鍵発行局）。PacketCable MTA のプロビジョニングに使用されます。

Kerberos
秘密鍵ネットワーク認証プロトコル。暗号化のための暗号アルゴリズムを選択し、認証のために中央集中鍵データベースを使用します。

M

MAC アドレス
LAN に接続するすべてのポートまたはデバイスに必要な、標準化されたデータリンク層アドレス。ネットワーク上の他のデバイスは、このアドレスを使用して、ネットワーク上の特定のポートの場所を割り出し、ルーティング テーブルとデータ構造を作成および更新します。MAC アドレスは 6 バイトの長さで、IEEE によって管理されています。ハードウェア アドレス、MAC 層アドレス、物理アドレスとも呼ばれます。*ネットワーク アドレス*と比較してください。

MSO
マルチプル システム オペレータ（multiple system operator）。複数のケーブル TV またはブロードバンド システムを運営する企業。

MTA
マルチメディア ターミナル アダプタ。ブロードバンド（PacketCable）ネットワークの顧客側の終端装置。

N

NAT
ネットワーク アドレス変換。グローバルに一意な IP アドレスの必要性を減らすメカニズム。NAT では、グローバルに一意でないアドレスをグローバルにルーティング可能なアドレス空間に変換することで、このようなアドレスを持つ組織をインターネットに接続できます。Network Address Translation とも言います。

NR
Cisco Network Registrar。ネットワーク ポリシーおよびサービス ポリシーに基づき、IP アドレス、構成パラメータ、DNS 名を DOCSIS ケーブル モデムおよび PC に提供するソフトウェア製品。

NTP
ネットワーク タイム プロトコル（Network Time Protocol）。ネットワークを通じてサーバ クロックを同期させるためのプロトコルです。

P

PacketCable
双方向ケーブル ネットワークを介した高度なリアルタイム マルチメディア サービスの配信に向けた、相互運用可能なインターフェイス仕様のための CableLabs イニシアティブ。ケーブル モデム インフラストラクチャ上に構築され、IP テレフォニー、マルチメディア会議、対話形式のゲーム、一般的なマルチメディア アプリケーションなど、広範囲のマルチメディア サービスを可能にします。

R

RDU
Regional Distribution Unit。BAC プロビジョニング システムのプライマリ サーバ。デバイス構成の生成を管理し、すべての API 要求を処理し、BAC システムを管理します。

T

TFTP Trivial File Transfer Protocol。あるコンピュータから別のコンピュータにネットワーク経由でファイルを転送できるようにする File Transfer Protocol（FTP; ファイル転送プロトコル）の簡易バージョン。

TLV Type-Length-Value。DOCSIS または PacketCable 設定ファイル内のタプル。

Type Length Value 「TLV」を参照。

U

uBr ユニバーサル ブロードバンド ルータ（Cisco 7246 または 7223 など）。DOCSIS CMTS を備えた Cisco ルータです。

V

VoIP VoIP は、IP ベースのデータ ネットワークによる通話呼および FAX 送信を行うための機能です。最適な QoS と優れた費用対効果を発揮します。

X

XGCP ネットワーク間でデータを受け渡す際に使用されるゲートウェイ コントロール プロトコル。これには、M（メディア）GCP および S（簡易）GCP が含まれています。

あ

アラート 問題をオペレータまたは管理者に通知する syslog または SNMP メッセージ。

う

ウォッチドッグ RDU、Tomcat、SNMP エージェントなどの BAC コンポーネント プロセスを監視、停止、起動、再起動するデーモン プロセス。

お

オプション、DHCP DHCP メッセージのオプション フィールドに保存された DHCP 設定パラメータおよびその他の制御情報。DHCP クライアントは、DHCP パケットでどのオプションを要求し送信するのかを判定します。Network Registrar は、オプション定義およびクライアントが属するオプション セットを作成できます。

か

監査ログ RDU データベースの大きな変更の概要が含まれているログ ファイル。システム デフォルト、テクノロジー デフォルト、サービス クラスの変更が含まれます。

き

キャッシング　前のトランザクションで学習した情報を後のトランザクションで処理するために使用する複製の形式。

共有秘密情報　2 台のサーバまたはデバイス間で安全な通信を行うために使用する文字列。

く

クライアント クラス　Network Registrar の機能で、共通のネットワークに接続されているユーザを区別して異なるサービスを提供します。クライアント クラスは、BAC DHCP 基準で使用され、異なる DHCP サービスをデバイスに提供します。

け

ケーブル モデム ターミネーション システム　「CMTS」を参照。

こ

構成の生成　デバイスに対する構成を RDU で生成し、それらの構成を DPE に配信するプロセス。構成命令は DPE によってキャッシュされ、CPE で実行する必要があるアクションについての情報が提供されます。

し

冗長性　インターネットワーキングでの、デバイス、サービス、接続などの複製。障害が発生した場合は、障害が発生したデバイス、サービス、接続の代わりに、冗長なデバイス、サービス、接続が機能を実行します。

シングル スタック　DOCSIS ケーブル モデムの動作モード。このモードで、モデムは、常に単一の IP アドレス タイプ（v4 または v6）で動作します。

せ

静的設定ファイル　これらのファイルはデバイスの設定ファイルとして使用します。たとえば、*gold.cm* と呼ばれる静的設定ファイルは、gold DOCSIS サービス クラスを特定します。BAC は、このファイル タイプをその他のバイナリ ファイルと同様に扱います。

設定ファイル　プロビジョニングするデバイスの構成パラメータが含まれているファイル。

選択タグ　Network Registrar スコープに関連付けられた選択タグ。スコープに関連付けられたクライアントおよびクライアント クラスを定義します。

そ

組織固有識別子（OUI）　VPN の所有者または ISP を特定するために IEEE が割り当てます。

た

帯域幅
ネットワーク信号で利用可能な最高周波数と最低周波数の差。あるネットワーク メディアまたはプロトコルの定格スループット キャパシティを表すこともあります。

タプル
プログラミング言語では、タプルは順序付けされた値のセット。データ型としてのタプルは、一般的に、あるプログラムから別のプログラムにパラメータ文字列を渡したり、リレーショナル データベースの値属性のセットを表すために使用されます。

て

データオーバーケーブル サービス インターフェイス仕様
「DOCSIS」を参照。

デュアル スタック
DOCSIS ケーブル モデムの動作モード。モデムは IPv4 と IPv6 の両方のアドレスによって同時に管理できます。

テンプレート ファイル
DOCSIS または PacketCable MTA オプションを含むテキスト ファイル。DOCSIS または PacketCable MTA サービス クラスとともに使用する場合、動的にファイルを生成します。

と

動的設定ファイル
動的に作成された設定ファイル。テンプレート ファイルを使用して、プロビジョニング プロセスにおいて柔軟性とセキュリティを向上させます。

ドメイン
DNS ネーミング階層ツリーの一部。組織のタイプや地理的条件に基づいたネットワークの一般的なグループ化を参照します。階層は、ルート、トップまたは第 1 レベル ドメインおよび第 2 レベル ドメインです。

ね

ネットワーク アドレス
物理的ではなく論理的なネットワーク デバイスを参照するネットワーク レイヤ アドレス。プロトコル アドレスとも言います。「*MAC アドレス*」と比較してください。

ネットワーク オペレータ
日常的にネットワークを監視および制御し、アラームの確認と対応、スループットの監視、新しい回線の構成、問題の解決などの作業を実行する人。「ネットワーク管理者」も参照。

ネットワーク管理者
ネットワークの運用、メンテナンス、および管理を担当する人。「ネットワーク オペレータ」も参照。

ネットワーク タイム プロトコル
「NTP」を参照。

は

パブリッシング
プロビジョニング情報を外部データストアにリアルタイムで提供するプロセス。データをデータストアに書き込むために、パブリッシング プラグインを開発する必要があります。

ふ

ブロードバンド
複数の独立した信号を 1 本のケーブルに多重化する転送システム。テレコミュニケーションの用語では、音声レベルのチャネル（4 kHz）を超える帯域幅のチャネルのことです。LAN の用語では、アナログ シグナリングを使用する同軸ケーブルのことです。

プロビジョニング API
オペレーティング システムにさまざまな機能を実行させるために、プログラムで使用できる一連の BAC 関数。

プロビジョニング グループ
ネットワーク トポロジまたは地理的条件に基づいて、関連付けられた DPE サーバおよび DHCP サーバの定義済みセットを持つデバイスのグループ。

ま

マルチプル サービス オペレータ
「MSO」を参照。

り

リース クエリー
このプロセスによって、リレー エージェントはリース（および予約）データを DHCP サーバから直接要求し、クライアント / サーバ トランザクションからこれらのデータを収集することができます。

リレー エージェント
2 つ以上のネットワークまたはネットワーク システムを接続するデバイス。DHCP では、DHCP サーバの IP ヘルパーであるバーチャル プライベート ネットワーク上にあるルータ。

れ

レルム
単一の Kerberos データベースおよび複数の鍵発行局によって管理される論理ネットワーク。

レルム名
慣例的に、レルム名はすべて大文字であり、インターネット ドメインと区別されます。「レルム」を参照。

INDEX

A

B

C

D

O

P

R

S

T

U

V

W

あ

い

う

え

お

か

き

け

こ

さ

し

せ

ち

つ

て

と

の

は

ふ

へ

む

ゆ

ら

り

れ

ろ