



Cisco CMTS MIB の使用

この章では、SNMP（簡易ネットワーク管理プロトコル）要求を使用し、Cisco CMTS ユニバーサルブロードバンドルータ上で次の作業を行うために必要なオブジェクトおよびMIB（管理情報ベース）について説明します。

- [ヒントおよび注意事項 \(p.A-2\)](#)
- [基本的なルータ情報の取得 \(p.A-4\)](#)
- [物理コンポーネントの管理 \(p.A-6\)](#)
- [SNMP トラップの生成 \(p.A-13\)](#)
- [Syslog メッセージの監視 \(p.A-17\)](#)
- [CM 情報の表示 \(p.A-19\)](#)
- [スペクトル管理の監視 \(p.A-24\)](#)
- [フラップリストの使用法 \(p.A-33\)](#)
- [加入者トラフィック管理機能の使用法 \(p.A-38\)](#)
- [使用状況ベースの課金 \(p.A-41\)](#)
- [Cisco Unique Device Identifier の識別 \(p.A-45\)](#)
- [DOCS-DSG-IF-MIB 検証の要件 \(p.A-46\)](#)

ヒントおよび注意事項

SNMP を使用して Cisco CMTS ルータを管理する場合の注意事項は、次のとおりです。

IF-MIB キャッシング

Cisco IOS Release 12.2(15)BC1 以降のリリースでは、Cisco CMTS ルータはキャッシュを実装し、CPU 使用率の急増を引き起こすことなく、ifTable インターフェイス カウンタの連続ポーリングを実行できるようになっています。このようなカウンタに対する SNMP 要求には、カウンタの最新の実行値ではなく、カウンタ キャッシュ メモリに最後に格納された値が返されます。この方式では SNMP 要求が行われたときに、Cisco IOS ソフトウェアで各ラインカードにポーリングを実行してカウンタを取得する必要がないので、パフォーマンスが向上します。

ifTable カウンタ キャッシュは約 10 秒間隔で更新されます。したがって、10 秒より短い間隔で ifTable インターフェイス カウンタを読み取るような SNMP 要求では、新しい値が返されない可能性があります。実行カウンタはインターフェイス上で発生した実際のトラフィックに応じて増分を続けますが、次の SNMP 要求が 10 秒以内に行われた場合、新しい値は反映されません。

SNMP ベースのカウンタおよび CLI ベースのカウンタ

SNMP 仕様では通常、システムの初期化時を除き、SNMP ベース カウンタについて消去が認められません。正常に稼働している間、カウンタは最大値に到達するまで増分を続け、最大値に達するとゼロから再び増分します。

SNMP コマンドを使用してルータを管理する場合は、次の点でこの動作を考慮する必要があります。

- 32 ビットカウンタ — 32 ビットカウンタは約 42 億に到達すると、ゼロに戻ります。したがって、フル稼働しているルータの場合、わずか数日でバイトカウンタやパケットカウンタがゼロに戻る可能性があります。パケットまたはその他のオブジェクトについて、正しいカウントを確実に維持するには、該当するカウンタのポーリングを定期的に行い、必ず直前の値を保存してください。現在の値から直前の値を差し引いて、2 つのカウンタの差がマイナスになった場合は、いったんゼロに戻ったということの意味します。

数週間または数か月単位でカウンタの合計を正確に算出するには、この期間にカウンタがゼロに戻った回数を追跡することも必要です。ゼロへの折り返しが検出されないということが 1 回以上にならないように、頻繁にカウンタのポーリングを実行してください。



ヒント

一部の SNMPv3 MIB では、同じオブジェクトの多くについて、32 ビットカウンタとともに 64 ビットカウンタが含まれるようになりました。選択する場合は、64 ビットカウンタを使用してください。64 ビットカウンタであれば通常、何か月または何年も、あるいはさらに長期にわたってゼロに戻らないからです。

- 指定したイベントからカウントする場合または期間限定でカウントする場合 — SNMP ベースカウンタは、ルータの電源がオンになったときにゼロからカウントを始め、折り返すまで増分を続けます。特定のイベント以後のパケットまたはその他のオブジェクトの数を追跡するには、イベント発生時点でカウンタの値を保存する必要があります。その後、新しいパケットカウントを取得するときに、カウンタの現在値と保存した値を比較します。
- CLI (コマンドライン インターフェイス) 値との比較 — **show** コマンドの多くには、カウンタをゼロにリセットする **clear** コマンドが関連付けられています。ただし、**clear** コマンドが作用するのは、CLI によって表示されるカウンタだけです。SNMP ベースのカウンタには無効です。また、多くの CLI ベース カウンタは、インターフェイスのリセットなど、特定の機能が実行さ

れるたびに自動的にリセットされます。したがって、CLI コマンドを使用して表示したカウンタと SNMP コマンドで表示したカウンタとは、通常、一致しません。CLI ベースのカウンタと SNMP ベースのカウンタを比較する場合は、この相違に注意してください。

Cisco uBR10012 ルータ上の冗長 PRE モジュール

Cisco IOS Release 12.2BC が稼働している Cisco uBR10012 ルータでは、SNMP コンフィギュレーション コマンドは CLI コマンドの場合と同様、スタンバイの冗長 Performance Routing Engine (PRE) モジュールに同期しません。したがって、切り替えが発生し、スタンバイ PRE モジュールがアクティブになっても、SNMP コマンドによる設定はアクティブになりません。ルータが最初の PRE モジュールに切り替えると、元の設定が復元されます。

重要な設定には CLI コマンドを使用し、スタートアップ コンフィギュレーションに保存して、切り替え時に確実にアクティブになるようにしてください。

基本的なルータ情報の取得

Cisco CMTS ルータの基本情報は、次の MIB オブジェクトから取得できます。

- [OLD-CISCO-CHASSIS-MIB \(p.A-4\)](#)
- [SNMPv2-MIB \(p.A-4\)](#)
- [ENTITY-MIB \(p.A-5\)](#)

OLD-CISCO-CHASSIS-MIB

OLD-CISCO-CHASSIS-MIB の次のオブジェクトは、ルータ シャーシのシリアル番号を保管するのに適切な場所です。テクニカル サポートに連絡するとき、ここから番号を簡単に取り出すことができます。

- `shassisId` — `snmp-server chassis-id` コマンドの定義に従って、シャーシのシリアル番号または ID 番号を提供します。通常、購入したサービス契約およびサービス レベルを調べる場合に使用します。このオブジェクトのデフォルトは空の文字列なので、`snmp-server chassis-id` コマンドを使用してオブジェクトの値を設定したあとにのみ、値を取り出すことができます。

```

csh% getmany -v2c 10.10.11.12 public chassisId

chassisId.0 = TBA06500113

```

SNMPv2-MIB

SNMPv2-MIB の次のオブジェクトは、ルータの基本情報、ソフトウェア情報、およびその他の実行情報を提供します。

- `sysDescr` — モデル番号、稼働している Cisco IOS ソフトウェアのバージョンなど、ルータに関する総合的な説明を提供します。次に例を示します。

```

csh% getmany -v2c 10.10.11.12 public sysDescr

sysDescr.0 = Cisco Internetwork Operating System Software
IOS (tm) 10000 Software (UBR10K-K8P6-M), Released Version 12.2(15)BC1
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Fri 23-Jan-04 23:56 by atifg

```

- `sysObjectID` — CISCO-PRODUCTS-MIB で定義されている、特定のモデル番号を提供します。次に例を示します。

```

csh% getmany -v2c 10.10.11.12 public sysObjectID

sysObjectID.0 = ciscoProducts.ciscoUBR10012

```

- `sysName` — `hostname` コマンドで割り当てられたホスト名を提供します。次に例を示します。

```

csh% getmany -v2c 10.10.11.12 public sysName

sysName.0 = UBR10012-Router

```

- `sysUpTime` — ルータが前回初期化されてからの経過時間を 100 分の 1 秒単位で提供します。次に例を示します。

```

csh% getmany -v2c 10.10.11.12 public sysUpTime

sysUpTime.0 = 138389875

```

- **sysContact** — **snmp-server contact** コマンドで入力された情報に従って、このルータの担当者または担当部門の名前、電話番号、またはその他の識別情報を提供します。次に例を示します。

```
csh% getmany -v2c 10.10.11.12 public sysContact
```

```
sysContact.0 = IT Support at 408-555-1212 or epage it-support
```

- **sysLocation** — **snmp-server location** コマンドで入力された情報に従って、ルータの位置情報を提供します。次に例を示します。

```
csh% getmany -v2c 10.10.11.12 public sysLocation
```

ENTITY-MIB

ENTITY-MIB の次のオブジェクトは、基本的なルータ ハードウェア情報を提供します。

- **entPhysicalDescr** — ルータを構成する個々のハードウェア コンポーネントに関する記述を提供します。次に、Cisco uBR7246VXR シャーシの一般的な記述例を示します。

```
csh% getnext -v2c 10.10.11.12 public entPhysicalDescr
```

```
entPhysicalDescr.1 = uBR7246VXR chassis, Hw Serial#: 65100, Hw Revision: A
```

- **entPhysicalHardwareRev** — 各コンポーネントにハードウェア リビジョンが存在し、なおかつそのコンポーネントでサポートされている場合に、そのリビジョン情報を提供します。次に例を示します。

```
csh% getnext -v2c 10.10.11.12 public entPhysicalHardwareRev
```

```
entPhysicalHardwareRev.1 = 1.1
```

- **entPhysicalSerialNum** — 各コンポーネントにシリアル番号が存在し、なおかつそのコンポーネントでサポートされている場合に、そのシリアル番号を提供します。次に例を示します。

```
csh% getnext -v2c 10.10.11.12 public entPhysicalSerialNum
```

```
entPhysicalSerialNum.1 = TBC06481339
```

- **entPhysicalModelName** — 各コンポーネントにモデル名が存在し、なおかつそのコンポーネントでサポートされている場合に、そのモデル名を提供します。次に例を示します。

```
csh% getnext -v2c 10.10.11.12 public entPhysicalModelName
```

```
entPhysicalModelName.1 = uBR7246VXR
```



ヒント

ENTITY-MIB の詳細および使用方法については、次のセクションを参照してください。

物理コンポーネントの管理

Cisco CMTS ルータは、ルータの物理コンポーネントを管理するため複数の MIB をサポートします。これらの MIB は次の機能を提供します。

- シャーシ内の物理エンティティを、各エンティティと他のすべてのエンティティとの関係について記述する包含ツリーに編成します。
- Field-Replaceable Unit (FRU; 現場交換可能ユニット) のステータスを監視および設定します。
- 物理ポートをそれぞれ対応するインターフェイスにマッピングします。
- 資産タギングのための資産情報を提供します。
- シャーシコンポーネントに関するファームウェアおよびソフトウェア情報を提供します。

以下では、各 MIB について説明するとともに、MIB を使用してルータ内のコンポーネントを追跡する手順について説明します。

- ENTITY-MIB (p.A-6)
- シスコ固有の MIB (p.A-7)
- インベントリ管理の実行 (p.A-7)



ヒント

ルータ シャーシのシリアル番号を調べるには、OLD-CISCO-CHASSIS-MIB から chassisId オブジェクトを取り出します。このオブジェクトのデフォルトは空の文字列なので、`snmp-server chassis-id` コマンドを使用してオブジェクトの値を設定したあとにのみ、値を取り出すことができます。

ENTITY-MIB

Cisco CMTS ルータは、標準の RFC 2737 で定義された ENTITY-MIB を使用して、ルータの物理コンポーネント (エンティティ) を管理します。エンティティには、カード、カード上のポート、カード上の主要サブシステム、シャーシスロット、FRU、またはルータに搭載されている他のあらゆる機器があります。

ENTITY-MIB では、個々のエンティティ間の関係を示す階層構造の包含ツリーを使用して、ルータの各エンティティを固有のものとして識別するオブジェクトのセットを定義します。他の MIB は ENTITY-MIB によって定義されたオブジェクトを使用し、各エンティティについて追加情報を提供できます。

ルータの物理エンティティを管理する場合、ENTITY-MIB で最も重要なオブジェクトは次のとおりです。

- entPhysicalTable — ルータ上の各物理コンポーネント (エンティティ) を記述します。このテーブルには、最上位エンティティ (シャーシ) の行エントリがあり、さらにシャーシ内のエンティティごとにエントリがあります。各エントリは、そのエントリの名前および説明、タイプおよびベンダー、さらにエンティティが包含ツリーのどこに収まるかどうかについて、説明を示します。
- entPhysicalIndex — 各エントリを固有のものとして識別します。この値は、そのシャーシ内の全機器にわたって、またすべての MIB にわたって固有であることが保証されるため、複数の MIB からのデータを特定のエントリに関連付けることができます。
- entAliasMappingTable — 各物理ポートの entPhysicalIndex 値を IF-MIB の ifTable 内の対応する ifIndex 値にマッピングします。これにより、特定のインターフェイスで特定のポートを速やかに識別できます。

Cisco IOS Release 12.2(15)BC2 以降のリリースでは、Cisco uBR-MC5X20S および Cisco uBR-MC5X20U ケーブルインターフェイス ラインカード上で仮想インターフェイスが設定されている場合、entAliasMappingTable は論理アップストリーム インターフェイスに対する

物理アップストリーム コネクタのマッピングも示します。このサポートにより、ケーブル インターフェイス ラインカードとそのポート間の親子関係も変更されました。詳細については、「[仮想インターフェイスのサポートの変更](#)」(p.3-75) を参照してください。

- **entPhysicalContainsTable** — 物理エンティティごとに、そのエンティティの子オブジェクトに対応する **entPhysicalIndex** 値を示します。これにより、ルータのコンテナ ツリーを簡単に作成できます。包含ツリーは、シャーシ上の物理エンティティ間の関係を示します。

通常、コンテナ ツリーは次のように編成されます。

- シャーシは最上位レベルであり、プロセッサ カードおよびシャーシ スロットが含まれます。
- シャーシ スロットには個々のカードおよび入出力コントローラ (搭載されている場合) が含まれます。
- ラインカードにはポート (インターフェイス) が含まれます。
- ケーブル インターフェイス ラインカードには、ダウンストリーム ポート (通称、CLC [ケーブル ラインカード]) が含まれます。Cisco IOS Release 12.2(15)BC1 およびそれ以前のリリースでは、各ダウンストリーム ポートに対応するアップストリーム ポートが含まれません。Cisco IOS Release 12.2(15)BC2 およびそれ以降のリリースでは、ダウンストリーム ポートおよびアップストリームはすべて、ケーブル インターフェイス ラインカードの子になります。

シスコ固有の MIB

Cisco CMTS ルータは、ENTITY-MIB 以外に次の MIB を使用して、ルータに搭載されている物理コンポーネントの補足情報を提供します。

- **CISCO-ENTITY-ASSET-MIB** — ENTITY-MIB の **entPhysicalTable** に示された物理エンティティに関する資産トラッキング情報 (ID PROM コンテンツ) が含まれています。この MIB は、物理エンティティに関するデバイス固有の情報 (発注時に使用する部品番号、シリアル番号、製造番号、ハードウェア、ソフトウェア、ファームウェア情報など) を提供します。
- **CISCO-ENTITY-FRU-CONTROL-MIB** — ENTITY-MIB の **entPhysicalTable** に示された FRU (電源装置、ラインカードなど) の管理ステータスおよび動作ステータスを監視したり設定したりするために使用するオブジェクトが含まれています。



(注) 現在のところ、CISCO-ENTITY-FRU-CONTROL-MIB はラインカードのみをサポートしています。

- **CISCO-ENTITY-VENDORTYPE-OID-MIB** — ルータ上のすべての物理エンティティに関する Object Identifier (OID; オブジェクト ID) が含まれています。
- **CISCO-ENVMON-MIB** — 環境センサ (電圧、温度、ファン、および電源装置) のステータス情報が含まれています。たとえば、この MIB はシャーシの内部および吸気口の温度を報告します。

インベントリ管理の実行

ENTITY-MIB は、ルータの物理コンポーネントに関するインベントリを収集するために必要なあらゆる情報を提供します。Cisco uBR7246VXR ルータを使用してこれを行う方法を次の手順で説明します。この例では、ルータに次のカードが搭載されています。

- 入出力スロット: デュアル ファスト イーサネット 入出力コントローラ (FastEthernet0/0 および FastEthernet0/1)
- スロット 1: ファスト イーサネット ラインカード (FastEthernet1/0 および FastEthernet1/1)
- スロット 4: Cisco uBR-MC28U
- スロット 6: Cisco uBR-MC16E

ENTITY-MIB の情報を集めて編成する手順は、次のとおりです。

- ステップ 1** すべての `entPhysicalDescr` オブジェクトを表示することで、物理エンティティのリストを収集します。次に例を示します。

```

entPhysicalDescr.1 = uBR7246VXR chassis, Hw Serial#: 65100, Hw Revision: A
entPhysicalDescr.2 = NPE 400 Card
entPhysicalDescr.3 = Chassis Slot
entPhysicalDescr.4 = I/O Dual Fast Ethernet Controller
entPhysicalDescr.5 = i82543 (Livengood)
entPhysicalDescr.6 = i82543 (Livengood)
entPhysicalDescr.7 = Chassis Slot
entPhysicalDescr.8 = Dual Port Fast Ethernet (RJ45)
entPhysicalDescr.9 = i82543 (Livengood)
entPhysicalDescr.10 = i82543 (Livengood)
entPhysicalDescr.11 = Chassis Slot
entPhysicalDescr.12 = Chassis Slot
entPhysicalDescr.13 = Chassis Slot
entPhysicalDescr.14 = MC28U_F_connector
entPhysicalDescr.15 = UBR7200 CLC
entPhysicalDescr.16 = UBR7200 CLC
entPhysicalDescr.17 = BCM3138 PHY
entPhysicalDescr.18 = BCM3138 PHY
entPhysicalDescr.19 = BCM3138 PHY
entPhysicalDescr.20 = BCM3138 PHY
entPhysicalDescr.21 = BCM3034 PHY
entPhysicalDescr.22 = BCM3138 PHY
entPhysicalDescr.23 = BCM3138 PHY
entPhysicalDescr.24 = BCM3138 PHY
entPhysicalDescr.25 = BCM3138 PHY
entPhysicalDescr.26 = BCM3034 PHY
entPhysicalDescr.27 = Chassis Slot
entPhysicalDescr.28 = Chassis Slot
entPhysicalDescr.29 = MC16E
entPhysicalDescr.30 = BCM3210 ASIC
entPhysicalDescr.31 = BCM3137 PHY
entPhysicalDescr.32 = BCM3137 PHY
entPhysicalDescr.33 = BCM3137 PHY
entPhysicalDescr.34 = BCM3137 PHY
entPhysicalDescr.35 = BCM3137 PHY
entPhysicalDescr.36 = BCM3137 PHY
entPhysicalDescr.37 = BCM3033 PHY

```

- ステップ 2** `entPhysicalVendorType`、`entPhysicalName`、および `entPhysicalClass` オブジェクトを収集することで、各 `entPhysicalDescr` オブジェクトの詳細情報を取得します。インデックス値を使用して、オブジェクトと対応する `entPhysicalDescr` オブジェクトを突き合わせます。表 A-1 に、この例で使用するオブジェクトの一般的な記述を示します。

表 A-1 `entPhysicalDescr` オブジェクトおよび記述例

インデックス番号	<code>entPhysicalDescr</code> ¹	<code>entPhysicalVendorType</code>	<code>entPhysicalName</code>	<code>entPhysicalClass</code>
1	uBR7246VXR chassis, Hw Serial#:012345, Hw Revision:A	cevChassisUbr7246Vxr		chassis(3)
2	NPE 400 Card	cevCpu7200Npe400		module(9)
3	Chassis Slot	cevContainerSlot		container(5)
4	I/O Dual Fast Ethernet Controller	cevC7xxxIo2FE		module(9)
5	i82543 (Livengood)	cevPortFEIP	FastEthernet0/0	port(10)
6	i82543 (Livengood)	cevPortFEIP	FastEthernet0/1	port(10)
7	Chassis Slot	cevContainerSlot		container(5)
8	Dual Port Fast Ethernet (RJ45)	cevPa2feTxI82543		module(9)
9	i82543 (Livengood)	cevPortFEIP	FastEthernet1/0	port(10)
10	i82543 (Livengood)	cevPortFEIP	FastEthernet1/1	port(10)
11	Chassis Slot	cevContainerSlot		container(5)
12	Chassis Slot	cevContainerSlot		container(5)
13	Chassis Slot	cevContainerSlot		container(5)
14	MC28U_F_connector	cevModuleUbrType		module(9)
15	UBR7200 CLC	cevPortRfMac	Cable4/0	port(10)

表 A-1 entPhysicalDescr オブジェクトおよび記述例 (続き)

インデックス番号	entPhysicalDescr ¹	entPhysicalVendorType	entPhysicalName	entPhysicalClass
16	UBR7200 CLC	cevPortRfMac	Cable4/1	port(10)
17	BCM3138 PHY	cevPortRfUs	Cable4/0-upstream0	port(10)
18	BCM3138 PHY	cevPortRfUs	Cable4/0-upstream1	port(10)
19	BCM3138 PHY	cevPortRfUs	Cable4/0-upstream2	port(10)
20	BCM3138 PHY	cevPortRfUs	Cable4/0-upstream3	port(10)
21	BCM3034 PHY	cevPortRfDs	Cable4/0-downstream	port(10)
22	BCM3138 PHY	cevPortRfUs	Cable4/1-upstream0	port(10)
23	BCM3138 PHY	cevPortRfUs	Cable4/1-upstream1	port(10)
24	BCM3138 PHY	cevPortRfUs	Cable4/1-upstream2	port(10)
25	BCM3138 PHY	cevPortRfUs	Cable4/1-upstream3	port(10)
26	BCM3034 PHY	cevPortRfDs	Cable4/1-downstream	port(10)
27	Chassis Slot	cevContainerSlot		container(5)
28	Chassis Slot	cevContainerSlot		container(5)
29	MC16E	cevUbrMc16e		module(9)
30	BCM3210 ASIC	cevPortRfMac	Cable6/0	port(10)
31	BCM3137 PHY	cevPortRfUs	Cable6/0-upstream0	port(10)
32	BCM3137 PHY	cevPortRfUs	Cable6/0-upstream1	port(10)
33	BCM3137 PHY	cevPortRfUs	Cable6/0-upstream2	port(10)
34	BCM3137 PHY	cevPortRfUs	Cable6/0-upstream3	port(10)
35	BCM3137 PHY	cevPortRfUs	Cable6/0-upstream4	port(10)
36	BCM3137 PHY	cevPortRfUs	Cable6/0-upstream5	port(10)
37	BCM3033 PHY	cevPortRfDs	Cable6/0-downstream	port(10)

1. インターフェイスは通常、インターフェイスのコネクタに使用されているチップセットで識別します。ケーブルインターフェイスではさらに、物理レイヤに接続するチップセットによってアップストリームを識別します。

ステップ 3 ルータの包含ツリーを作成するには、entPhysicalDescr オブジェクトごとに EntPhysicalContainedIn オブジェクトを収集します。EntPhysicalContainedIn の値は、対応する entPhysicalDescr デバイスの親 (または「コンテナ」) のインデックス番号です。

次に、この例で使用しているオブジェクトの entPhysicalContainedIn 値を示します。

```

entPhysicalContainedIn.1 = 0
entPhysicalContainedIn.2 = 1
entPhysicalContainedIn.3 = 1
entPhysicalContainedIn.4 = 3
entPhysicalContainedIn.5 = 4
entPhysicalContainedIn.6 = 4
entPhysicalContainedIn.7 = 1
entPhysicalContainedIn.8 = 7
entPhysicalContainedIn.9 = 8
entPhysicalContainedIn.10 = 8
entPhysicalContainedIn.11 = 1
entPhysicalContainedIn.12 = 1
entPhysicalContainedIn.13 = 1
entPhysicalContainedIn.14 = 13
entPhysicalContainedIn.15 = 14
entPhysicalContainedIn.16 = 14
entPhysicalContainedIn.17 = 15
entPhysicalContainedIn.18 = 15
entPhysicalContainedIn.19 = 15

entPhysicalContainedIn.20 = 15
entPhysicalContainedIn.21 = 15
entPhysicalContainedIn.22 = 16
entPhysicalContainedIn.23 = 16
entPhysicalContainedIn.24 = 16
entPhysicalContainedIn.25 = 16
entPhysicalContainedIn.26 = 16
entPhysicalContainedIn.27 = 1
entPhysicalContainedIn.28 = 1
entPhysicalContainedIn.29 = 28
entPhysicalContainedIn.30 = 29
entPhysicalContainedIn.31 = 30
entPhysicalContainedIn.32 = 30
entPhysicalContainedIn.33 = 30
entPhysicalContainedIn.34 = 30
entPhysicalContainedIn.35 = 30
entPhysicalContainedIn.36 = 30
entPhysicalContainedIn.37 = 30

```

表 A-2 に、この例で使用している entPhysicalDescr オブジェクトの親コンテナを示します。

表 A-2 EntPhysicalContainedIn および entPhysicalDescr の関係

#	entPhysicalDescr	格納先	entPhysicalContainedInValue および親コンテナ
1	uBR7246VXR chassis, Hw Serial#:65100, Hw Revision:A		0、最上位レベル、親コンテナなし
2	NPE 400 Card		1、Chassis
3	Chassis Slot		1、Chassis
4	I/O Dual Fast Ethernet Controller		3、Chassis Slot
5	i82543 (Livengood)		4、I/O Dual Fast Ethernet
6	i82543 (Livengood)		4、I/O Dual Fast Ethernet
7	Chassis Slot		1、Chassis
8	Dual Port Fast Ethernet (RJ45)		7、Chassis Slot
9	i82543 (Livengood)		8、Dual Port Fast Ethernet
10	i82543 (Livengood)		8、Dual Port Fast Ethernet
11	Chassis Slot		1、Chassis
12	Chassis Slot		1、Chassis
13	Chassis Slot		1、Chassis
14	MC28U_F_connector		13、Chassis Slot
15	UBR7200 CLC		14、MC28U_F connector
16	UBR7200 CLC		14、MC28U_F connector
17	BCM3138 PHY		15、UBR7200 CLC
18	BCM3138 PHY		15、UBR7200 CLC
19	BCM3138 PHY		15、UBR7200 CLC
20	BCM3138 PHY		15、UBR7200 CLC
21	BCM3034 PHY		15、UBR7200 CLC
22	BCM3138 PHY		16、UBR7200 CLC
23	BCM3138 PHY		16、UBR7200 CLC
24	BCM3138 PHY		16、UBR7200 CLC

表 A-2 EntPhysicalContainedIn および entPhysicalDescr の関係 (続き)

#	entPhysicalDescr	格納先	entPhysicalContainedInValue および親コンテナ
25	BCM3138 PHY		16、UBR7200 CLC
26	BCM3034 PHY		16、UBR7200 CLC
27	Chassis Slot		1、Chassis
28	Chassis Slot		1、Chassis
29	MC16E		28、Chassis Slot
30	BCM3210 ASIC		29、MC16E
31	BCM3137 PHY		30、BCM3210 ASIC
32	BCM3137 PHY		30、BCM3210 ASIC
33	BCM3137 PHY		30、BCM3210 ASIC
34	BCM3137 PHY		30、BCM3210 ASIC
35	BCM3137 PHY		30、BCM3210 ASIC
36	BCM3137 PHY		30、BCM3210 ASIC
37	BCM3033 PHY		30、BCM3210 ASIC

ステップ 4 (任意) 複数のラインカード スロット (シャーシ スロット) が含まれるルータのように、親オブジェクトにオブジェクト タイプが同じ子が複数含まれている場合は、entPhysicalParentRelPos オブジェクトを使用して、子オブジェクトを正しい順序に並べます。entPhysicalParentRelPos オブジェクトには、子オブジェクトの順番を示す整数が含まれます。この整数は通常、0 から始まるので、物理オブジェクトの実際の番号と一致します (スロット 0 には 0 の entPhysicalParentRelPos 値、スロット 1 には 1 の entPhysicalParentRelPos 値、以下同様)。



(注) entPhysicalParentRelPos に -1 が含まれている場合、そのオブジェクトは他のオブジェクトの間に識別可能な関係はありません。

表 A-3 に、対応する entPhysicalParentRelPos 値を使用することによって、シャーシ スロットを参照する entPhysicalDescr オブジェクトを物理順序に合わせて並べる方法を示します。たとえば、entPhysicalDescr.13 の entPhysicalParentRelPos 値である 4 は、このスロットがルータ シャーシのスロット 4/0 であることを意味します。

表 A-3 entPhysicalParentRelPos を使用して entPhysicalDescr オブジェクトを順序付ける方法

#	entPhysicalDescr	entPhysical ContainedIn	entPhysicalParent RelPos	物理スロット番号
1	uBR7246VXR chassis	0 = TopLevel	-1	適用外
2	NPE 400 Card	1 = Chassis	-1	なし
3	Chassis Slot	1 = Chassis	0	スロット 0/0
7	Chassis Slot	1 = Chassis	1	スロット 1/0
11	Chassis Slot	1 = Chassis	2	スロット 2/0
12	Chassis Slot	1 = Chassis	3	スロット 3/0
13	Chassis Slot	1 = Chassis	4	スロット 4/0
27	Chassis Slot	1 = Chassis	5	スロット 5/0
28	Chassis Slot	1 = Chassis	6	スロット 6/0

ステップ 5 (任意) IF-MIB で定義され、他の MIB で使用される ifIndex に物理インターフェイスをマッピングし、論理インターフェイスを固有のものとして識別するには、entAliasMappingIdentifier オブジェクトを使用します。ケーブル インターフェイス ラインカード上で仮想インターフェイスも設定されている場合、このテーブルにアップストリームの物理コネクタとその論理インターフェイス間のマッピングが示されます。

次に、この例で使用しているルータの entAliasMappingIdentifier 値を示します。この例では、entPhysicalDescr.5 (表 A-1 では FastEthernet0/0 インターフェイスとして識別) が 1 という ifIndex 値に対応しています。

```
entAliasMappingIdentifier.5.0 = ifIndex.1
entAliasMappingIdentifier.6.0 = ifIndex.2
entAliasMappingIdentifier.9.0 = ifIndex.3
entAliasMappingIdentifier.10.0 = ifIndex.4
entAliasMappingIdentifier.15.0 = ifIndex.5
entAliasMappingIdentifier.16.0 = ifIndex.6
entAliasMappingIdentifier.17.0 = ifIndex.8
entAliasMappingIdentifier.18.0 = ifIndex.9
entAliasMappingIdentifier.19.0 = ifIndex.10
entAliasMappingIdentifier.20.0 = ifIndex.11
entAliasMappingIdentifier.21.0 = ifIndex.12
entAliasMappingIdentifier.22.0 = ifIndex.13
entAliasMappingIdentifier.23.0 = ifIndex.14
entAliasMappingIdentifier.24.0 = ifIndex.15
entAliasMappingIdentifier.25.0 = ifIndex.16
entAliasMappingIdentifier.26.0 = ifIndex.17
entAliasMappingIdentifier.30.0 = ifIndex.7
entAliasMappingIdentifier.31.0 = ifIndex.18
entAliasMappingIdentifier.32.0 = ifIndex.19
entAliasMappingIdentifier.33.0 = ifIndex.20
entAliasMappingIdentifier.34.0 = ifIndex.21
entAliasMappingIdentifier.35.0 = ifIndex.22
entAliasMappingIdentifier.36.0 = ifIndex.23
entAliasMappingIdentifier.37.0 = ifIndex.24
```

SNMP トラップの生成

ここでは、ルータ上で特定のイベントまたは条件が発生すると SNMP トラップが生成されるように、Cisco CMTS ルータを設定する方法について説明します。SNMP コマンドを使用して SNMP トラップを生成するようにルータを設定するには、次の手順で、トラップを受信するターゲットホストを1つまたは複数定義する必要があります。



ヒント

CLI を使用して、ルータ上でのトラップ生成をイネーブルにして設定することもできます。CLI の使用方法については、「[通知のイネーブル化](#)」(p.4-3) を参照してください。

ステップ 1 SNMP-TARGET-MIB で定義された `snmpTargetAddrTable` に、トラップを受信するホストごとに1つエントリを作成します。各エントリには次のオブジェクトを含めます。

- `snmpTargetAddrName` — このホストを識別するための固有の文字列（最大 32 文字）
- `snmpTargetAddrTDomain` — このホストにトラップを配信するときに使用する TCP/IP トランスポート サービス。通常は `snmpUDPDomain`
- `snmpTargetAddrTAddress` — ホストのトランスポートアドレス。通常、ホストの 4 バイト IP アドレスにトラップの送信先となる 2 バイトの UDP ポート番号を加えて形成された、6 オクテット値
- `snmpTargetAddrTimeout` — Cisco CMTS ルータがホストからの応答（あれば）を待機する最大時間（100 分の 1 秒単位）。デフォルトは 1500（15 秒）です。
- `snmpTargetAddrRetryCount` — タイムアウトまでに応答を受信しなかった場合、Cisco CMTS ルータがトラップを再送信するデフォルトの回数。デフォルトの再試行値は 3 です。
- `snmpTargetAddrTagList` — 所定のターゲットホストに対応するタグ（後ろに定義）のリスト。ホストのタグ値が `snmpNotifyTag` 値と一致しなかった場合、ホストは対応する `snmpNotifyType` によって定義されたタイプの通知を受信します。
- `snmpTargetAddrParams` — トラップの生成に使用するパラメータを定義した、`snmpTargetParamsTable` のエントリを表す任意の文字列（最大 32 文字）
- `snmpTargetAddrStorageType` — この行エントリに使用するストレージのタイプ。 `volatile(2)`、`nonVolatile(3)`、`permanent(4)`、または `readOnly(5)` です。デフォルトは `nonVolatile(4)` です。
- `snmpTargetAddrRowStatus` — この行エントリを作成するには、`createAndGo(4)` または `createAndWait(5)` に設定する必要があります。このオブジェクトは、行の他のすべてのエントリを設定したあとで設定しなければなりません。

ステップ 2 SNMP-TARGET-MIB で定義された `snmpTargetParamsTable` にエントリを作成し、SNMP 通知の生成時にルータに使用させる SNMP パラメータを定義します。各エントリには次のオブジェクトを含めます。

- `snmpTargetParamsName` — このエントリを定義する固有の文字列（最大 32 文字）。`snmpTargetAddrParams` でもこの文字列を使用して、所定のホストにトラップを送信するときに使用するパラメータを定義します。
- `snmpTargetParamsMPModel` — このトラップを送信するときに使用する SNMP のバージョン。0=SNMPv1、1=SNMPv2c、および 3=SNMPv3 です。
- `snmpTargetParamsSecurityModel` — このトラップを送信するときに使用する SNMP セキュリティのバージョン。0=SNMPv1、1=SNMPv2c、および 3=SNMPv3 です。
- `snmpTargetParamsSecurityName` — トラップ送信時に Cisco CMTS ルータを識別するための文字列（最大 32 文字）

- `snmpTargetParamsSecurityLevel` — トラップを送信するとき使用するセキュリティのタイプ。`noAuthNoPriv(1)`、`authNoPriv(2)`、および `authPriv(3)` です。
- `snmpTargetParamsStorageType` — この行エントリに使用するストレージのタイプ。`volatile(2)`、`nonVolatile(3)`、`permanent(4)`、または `readOnly(5)` です。デフォルトは `nonVolatile(4)` です。
- `snmpTargetParamsRowStatus` — この行エントリを作成するには、`createAndGo(4)` または `createAndWait(5)` に設定する必要があります。このオブジェクトは、行の他のすべてのエントリを設定したあとで設定しなければなりません。

ステップ 3 SNMP-NOTIFICATION-MIB で定義された `snmpNotifyTable` にエントリを作成します。このテーブルの各行には、次のオブジェクトを含め、トラップを受信する一連のホストターゲットを定義します。

- `snmpNotifyName` — この行エントリを識別するための固有の文字列（最大 32 文字）
- `snmpNotifyTag` — トラップを受信する一連のホストを識別する任意の文字列（最大 255 文字）。このタグ値と `snmpTargetAddrTagList` オブジェクトを照合することによって、どのホストがどのトラップを受信するかが決定されます。
- `snmpNotifyType` — 設定するトラップのタイプを定義します。`trap(1)` または `inform(2)` です。デフォルトは `trap(1)` です。
- `snmpNotifyStorageType` — この行エントリに使用するストレージのタイプ。`volatile(2)`、`nonVolatile(3)`、`permanent(4)`、または `readOnly(5)` です。デフォルトは `nonVolatile(4)` です。
- `snmpNotifyRowStatus` — この行エントリを作成するには、`createAndGo(4)` または `createAndWait(5)` に設定する必要があります。このオブジェクトは、行の他のすべてのエントリを設定したあとで設定しなければなりません。

ステップ 4 SNMP-NOTIFICATION-MIB で定義された `snmpNotifyFilterProfileTable` および `snmpNotifyFilterTable` に、任意で行を作成します。これらのテーブルでは、ルータが特定のホストに送信する通知のタイプを制限する目的で、通知フィルタを作成します。

ステップ 5 任意で、トラップおよび通知の送信をイネーブルにします。通常、他の MIB には、機能固有のトラップをイネーブルまたはディセーブルにする、独自のオブジェクト `NOTIFICATION-TYPE` が組み込まれています。このような通知オブジェクトでも、発生したイベント固有の情報を格納し、各トラップとともに送信される変数バインドを定義します。

たとえば、`CISCO-CABLE-SPECTRUM-MIB` に組み込まれている `ccsHoppingNotification` オブジェクトは、周波数ホップが発生したとき、またはチャネル幅や変調プロファイルが変更されたときに送信されるトラップをイネーブルまたはディセーブルにします。このトラップは該当する場合に、ホップ時点でのアップストリームの状態、新旧の中心周波数、新旧のチャネル幅、および新旧の変調プロファイルを提供します。

CLI コマンドを使用してさまざまな通知およびトラップをイネーブルにすることもできます。表 A-4 に、一般的なトラップの一部を CLI でイネーブルにする方法、各トラップが生成される状況とともに示します。

表 A-4 一般的な通知およびトラップ

トラップのタイプ	イネーブルにする コンフィギュレーション コマンド	説明
設定の変更	<code>snmp-server enable traps entity</code>	<p>ENTITY トラップがイネーブルになっている場合、ルータは ENTITY-MIB の次のテーブルのいずれかで情報が変更されたときに、<code>entConfigChange</code> トラップを生成します。</p> <ul style="list-style-type: none"> • <code>entPhysicalTable</code> • <code>entAliasMappingTable</code> • <code>entPhysicalContainsTable</code> <p> (注) SNMP マネージャでも定期的に <code>entLastChangeTime</code> オブジェクトへのポーリングを実行し、スロットリングまたは伝送損失が原因でトラップが失われていないかどうかを調べます。</p>
環境変化	<code>snmp-server enable traps envmon</code>	<p>ENVMON トラップがイネーブルの場合、ルータは (CISCO-ENVMON-MIB で定義されている) 次のトラップを生成し、環境に関する潜在的な問題を伝えます。</p> <ul style="list-style-type: none"> • <code>ciscoEnvMonShutdownNotification</code> — ルータがシャットダウンする直前に送信されます。 • <code>ciscoEnvMonTemperatureNotification</code> — 温度が正常範囲を超えたときに送信されます。 • <code>ciscoEnvMonFanNotification</code> — ファンが故障したときに送信されます。 • <code>ciscoEnvMonRedundantSupplyNotification</code> — 冗長 PEM (パワー エントリ モジュール) が故障したときに送信されます。
FRU ステータスの 変更	<code>snmp-server enable traps fru-ctrl</code>	<p>Field-Replaceable Unit (FRU) トラップがイネーブルの場合、ルータは (CISCO-ENTITY-FRU-CONTROL-MIB で定義されている) 次のトラップを生成して、FRU の変更をホストに通知します。</p> <ul style="list-style-type: none"> • <code>cefcModuleStatusChange</code> — 変更された FRU の動作ステータス (<code>cefcModuleOperStatus</code>)。 • <code>cefcFRUInserted</code> — FRU がシャーシに挿入されました。 • <code>cefcFRURemoved</code> — FRU がシャーシから取り外されました。
アラームのアサート またはクリア	<code>snmp-server enable traps alarms</code>	<p>ALARM トラップがイネーブルの場合、ENTITY-MIB の <code>entPhysicalTable</code> で定義されている物理エンティティに対して、アラームがアサートまたはクリアされるたびに、ルータはトラップを生成します。</p>

■ SNMP トラップの生成

表 A-4 一般的な通知およびトラップ (続き)

トラップのタイプ	イネーブルにする コンフィギュレーション コマンド	説明
Syslog メッセージの 生成	<code>snmp-server enable traps syslog</code>	<p>デフォルトでは、Cisco CMTS ルータはアラームがアサートまたはクリアされるたびに Syslog メッセージを記録します。さらに、Syslog メッセージが記録されたときに別のトラップを生成する場合には、<code>clogNotificationsEnabled</code> object を <code>true(1)</code> に設定します。</p> <p>CISCO-SYSLOG-MIB に格納し、それについての通知を生成する Syslog メッセージの最大の重大度に CISCO-SYSLOG-MIB の <code>clogMaxSeverity</code> オブジェクトを設定します。デフォルトは 5 (警告) です。この場合、重大度 1～5 の Syslog メッセージが MIB によって処理されます。</p>

Syslog メッセージの監視

CISCO-SYSLOG-MIB では、正常な稼働中に Cisco CMTS ルータによって生成される Syslog メッセージの格納先となる、一連のオブジェクトを定義します。この MIB の定期的ポーリングによって、生成された Syslog メッセージのリストを取得できます。

メッセージ テーブル オブジェクト

Syslog メッセージはイネーブルの場合、`clogHistoryTable` のエントリとして保管されます。各 `clogHistoryEntry` には、保管されるメッセージごとに、次のオブジェクトが含まれます。

- `clogHistIndex` — テーブルに保管された個々の Syslog メッセージを固有のものとして識別するためのインデックス番号。このインデックスは、最大値に到達するまで連続して増加する 32 ビット値です。最大値に達すると、0 に戻ります。
- `clogHistFacility` — Syslog メッセージのファシリティ識別子 (最大 20 文字)
- `clogHistSeverity` — `SyslogSeverity` のテキスト規則によって定義された、Syslog メッセージの重大度。範囲は 1 (緊急) ~ 8 (デバッグ) です。



(注) `SyslogSeverity` オブジェクトおよび `clogHistSeverity` オブジェクトでは、重大度の値として、実際の Syslog メッセージで使用されるより 1 だけ大きい値を使用します。たとえば、エラーの Syslog メッセージの重大度は 3 ですが、`SyslogSeverity` ではエラーメッセージに 4 を使用します。

- `clogHistMsgName` — この Syslog メッセージのニーモニック (最大 30 文字)。30 文字を超えるニーモニックは、29 文字に切り詰められ、メッセージの末尾にアスタリスク (*) を加えて、切り詰められたことを表します。
- `clogHistMsgText` — コンソールおよび Syslog ログに表示される、実際の Syslog メッセージテキスト (最大 255 文字)。255 文字を超えるメッセージは、254 文字に切り詰められ、メッセージの末尾にアスタリスク (*) を加えて、切り詰められたことを表します。
- `clogHistTimestamp` — Syslog メッセージが生成されたときの `sysUpTime` に関するタイムスタンプ

制御オブジェクト

CISCO-SYSLOG-MIB の次のオブジェクトは、`clogHistoryTable` に保管するメッセージの数およびタイプを制御します。

- `clogMaxSeverity` — この MIB が処理する Syslog メッセージの最大重大度。デフォルトは 5 (警告) です。この場合、重大度 1 ~ 5 の Syslog メッセージが MIB によって処理されます。
- `clogMsgIgnores` — `clogMaxSeverity` で指定された重大度より大きいことが原因で無視された Syslog メッセージの数
- `clogMsgDrops` — リソース不足が原因で廃棄され、`clogHistoryTable` に保管されなかった Syslog メッセージの数
- `clogHistTableMaxLength` — `clogHistoryTable` に保存できる Syslog メッセージの最大数。テーブルが満杯になると、新しい Syslog メッセージが生成された時点で、スペースを作るためにテーブルで最も古いメッセージが削除されます。有効な範囲は 0 ~ 500、デフォルトは 1 です。
- `clogHistMsgsFlushed` — 新しいエントリ用のスペースを確保するために、`clogHistoryTable` から削除されたエントリの数。このオブジェクトが増え続ける場合は、テーブルサイズを増やすか (`clogHistTableMaxLength`)、またはテーブルポーリングの頻度を高める必要があります。

Syslog 通知

Syslog メッセージの生成時に SNMP 通知が生成されるように、Cisco CMTS ルータを設定できます。この通知では、Syslog メッセージを識別する次のオブジェクトが含まれた、`clogMessageGenerated` オブジェクトが送信されます。

- `clogHistFacility`
- `clogHistSeverity`
- `clogHistMsgName`
- `clogHistMsgText`
- `clogHistTimestamp`

CLI コマンドを使用して Syslog 通知をイネーブルにするには、グローバル コンフィギュレーション モードで次のコマンドを入力します。

```
snmp-server enable traps syslog
```

SNMP コマンドを使用して Syslog 通知をイネーブルにする場合は、`clogNotificationsEnabled` オブジェクトを `true(1)` に設定します。さらに `clogNotificationsSent` オブジェクトに、送信された `clogMessageGenerated` 通知の数が含まれます。

例

次に、多くの UNIX システムで標準となっている SNMP ユーティリティを使用した場合の、CISCO-SYSLOG-MIB の一般的な出力例を示します。このルータではデフォルトの設定を使用しているため、`clogHistoryTable` に保管される Syslog メッセージは 1 つだけです。テーブルには現在、インデックスが 25 のエントリがあります。それまでの 24 のメッセージはテーブルからすでにフラッシュされていることを `clogHistMsgsFlushed` が示しています。

```
csh% getmany -v2c 10.10.11.12 public ciscoSyslogMIB

clogNotificationsSent.0 = 0
clogNotificationsEnabled.0 = false(2)
clogMaxSeverity.0 = warning(5)
clogMsgIgnores.0 = 199
clogMsgDrops.0 = 0
clogHistTableMaxLength.0 = 1
clogHistMsgsFlushed.0 = 24
clogHistFacility.25 = UBR7200
clogHistSeverity.25 = error(4)
clogHistMsgName.25 = AUTH_REJECT_PERMANENT_AUTHORI*
clogHistMsgText.25 = <132>CMTS [DOCSIS]:<66030108> Auth Reject - Permanent
Authorization Failure . CM Mac Addr <000C.AB01.CD89>
clogHistTimestamp.25 = 4452551

csh%
```

CM 情報の表示

ここでは、1 つまたはすべての Cable Modem (CM) の情報を取得する方法について説明します。

- 現在の CM ステータスの表示 (p.A-19)
- バーストおよびステーションメンテナンス インターバルに関する情報の表示 (p.A-20)
- 削除されたサービス フローのログおよび表示 (p.A-21)

現在の CM ステータスの表示

1 つまたは複数の CM について、現在のステータスを表示するには、DOCS-IF-MIB で定義された docsIfCmtsCmStatusTable の docsIfCmtsCmStatusValue オブジェクトを表示します。このオブジェクトには、各 CM について、次のステートが含まれています。

- other(1) — オフラインなど、下記以外のあらゆるステート
- ranging(2) — CM は Cisco CMTS に初期レンジング要求を送信しましたが、レンジング プロセスはまだ完了していません。
- rangingAborted(3) — Cisco CMTS は CM にレンジング中止メッセージを送信しました。レンジング プロセスのやり直しが必要です。
- rangingComplete(4) — Cisco CMTS は CM にレンジング完了メッセージを送信しました。登録プロセスに進むことができます。
- ipComplete(5) — CM が DHCP ブロードキャストを送信し、Cisco CMTS が CM に、割り当てられた IP アドレスとともに DHCP 応答を転送しました。
- registrationComplete(6) — Cisco CMTS は CM に登録応答メッセージを送信しました、これは CM が登録プロセスを完了し、オンラインになって、接続 CPE 装置からのトラフィックを転送できることを伝えるメッセージです。
- accessDenied(7) — Cisco CMTS は CM に登録中止メッセージを送信しました。これはプロビジョニング システムが CM および接続 CPE 装置をオンラインにさせなかったことを伝えるメッセージです。

1 つ以上の CM について、現在のステータスを表示する手順は次のとおりです。

ステップ 1 docsIfCmtsCmStatusTable の docsIfCmtsCmStatusMacAddress オブジェクトに対するポーリングによって、既知の CM の MAC アドレスを取得します。

```
csh% getmany -v2c 10.10.17.91 public docsIfCmtsCmStatusMacAddress

docsIfCmtsCmStatusMacAddress.671745 = 00 0a ff 01 44 5e
docsIfCmtsCmStatusMacAddress.671746 = 00 0b fe 01 18 5e
docsIfCmtsCmStatusMacAddress.671747 = 00 0c fd 21 bb 54
docsIfCmtsCmStatusMacAddress.671748 = 00 0d fc 89 5f a9
docsIfCmtsCmStatusMacAddress.671749 = 00 0e fb 89 6b fd
docsIfCmtsCmStatusMacAddress.671750 = 00 0f fa 89 5c 6d
docsIfCmtsCmStatusMacAddress.671751 = 00 00 f0 89 5d 35

csh%
```

ステップ 2 docsIfCmtsCmStatusValue オブジェクトに対するポーリングによって、各 CM の現在のステータスを取得します。

```
csh% getmany -v2c 10.10.17.91 public docsIfCmtsCmStatusValue

docsIfCmtsCmStatusValue.671745 = registrationComplete (6)
docsIfCmtsCmStatusValue.671746 = registrationComplete (6)
docsIfCmtsCmStatusValue.671747 = registrationComplete (6)
docsIfCmtsCmStatusValue.671748 = registrationComplete (6)
docsIfCmtsCmStatusValue.671749 = accessDenied (7)
docsIfCmtsCmStatusValue.671750 = registrationComplete (6)
docsIfCmtsCmStatusValue.671751 = registrationComplete (6)

csh%
```

ステップ 3 docsIfCmtsCmStatusValue オブジェクトおよび docsIfCmtsCmStatusMacAddress オブジェクトのインデックス値を使用して、特定の CM に関する現在のステータスを調べます。たとえば、この例で表示される出力は、MAC アドレスが 00.0E.FB.89.6B.FD で、インデックスが 671749 の CM に現在、accessDenied(7) のステータスが設定されていることを示しています。

```
docsIfCmtsCmStatusMacAddress.671749 = 00 0e fb 89 6b fd
...
docsIfCmtsCmStatusValue.671749 = accessDenied (7)
```

バーストおよびステーション メンテナンス インターバルに関する情報の表示

アップストリームで使用するバースト インターバルおよびステーション メンテナンス インターバルの情報を表示するには、CISCO-DOCS-EXT-MIB の cdxUpInfoElemStatsIEType オブジェクトに対してポーリングを実行します。このオブジェクトは、アップストリームの ifDescr 値をインデックスとした 6 つのエントリからなるシーケンスです。

- cdxUpInfoElemStatsIEType.upstream-ifDescr.1 — 帯域要求で使用する要求バースト (reqIE) で現在使用しているミニスロットの数を表示します。
- cdxUpInfoElemStatsIEType.upstream-ifDescr.2 — 帯域要求またはショート データ パケット要求で使用する要求 / データ バースト (reqOrDataIE) で、現在使用しているミニスロットの数を表示します。
- cdxUpInfoElemStatsIEType.upstream-ifDescr.3 — オンラインにする新しい CM 用に確保した、初期メンテナンス バースト (initMtnIE) で、現在使用しているミニスロットの数を表示します。
- cdxUpInfoElemStatsIEType.upstream-ifDescr.4 — キープアライブおよびネットワーク メンテナンス メッセージで使用されるステーション メンテナンス バースト (stnMtnIE) で、現在使用しているミニスロットの数を表示します。
- cdxUpInfoElemStatsIEType.upstream-ifDescr.5 — ショート データ認可で使用されるショート データ認可バースト (shortGrantIE) で、現在使用しているミニスロットの数を表示します。
- cdxUpInfoElemStatsIEType.upstream-ifDescr.6 — 大型データ要求で使用するロング データ認可バースト (longGrantIE) で、現在使用しているミニスロットの数を表示します。

この情報を取得する手順は、次のとおりです。

- ステップ 1** 所定のアップストリームの ifIndex を取得します。それには ifDescr に対して GET 要求を行います。次に、Cisco uBR7246VXR ルータの出力例を示します。このルータにはファストイーサネットポートアダプタおよび Cisco uBR-MC16C ケーブルインターフェイスカードが 1 つずつ搭載されています。

```
csh% getmany -v2c 10.10.10.13 public ifDescr

ifDescr.1 = FastEthernet0/0
ifDescr.2 = FastEthernet0/1
ifDescr.3 = FastEthernet1/0
ifDescr.4 = FastEthernet1/1
ifDescr.5 = Cable4/0
ifDescr.8 = Cable4/0-upstream0
ifDescr.9 = Cable4/0-upstream1
ifDescr.10 = Cable4/0-upstream2
ifDescr.11 = Cable4/0-upstream3
ifDescr.12 = Cable4/0-upstream4
ifDescr.13 = Cable4/0-upstream5

csh%
```

- ステップ 2** 所定のアップストリームの ifIndex を 6 つの cdxUpInfoElemStatsIEType オブジェクトそれぞれに対するインデックスとして使用します。次の例では、Cisco uBR-MC16C カード上のアップストリーム 0 に対応する ifDescr を使用しています。

```
csh% getmany -v2c 10.10.10.13 public cdxUpInfoElemStatsIEType.8

cdxUpInfoElemStatsIEType.8.reqIE = 76826109
cdxUpInfoElemStatsIEType.8.reqOrDataIE = 0
cdxUpInfoElemStatsIEType.8.initMtnIE = 494562
cdxUpInfoElemStatsIEType.8.stnMtnIE = 47447
cdxUpInfoElemStatsIEType.8.shortGrantIE = 242
cdxUpInfoElemStatsIEType.8.longGrantIE = 29116

csh%
```



(注) Cisco CMTS は要求 / データ バーストをサポートしますが、使用することはないので、reqOrDataIE タイプの出力は常に 0 です。

削除されたサービス フローのログおよび表示

DOCSIS 2.0 の仕様により、CMTS は削除された DOCSIS 1.1 または DOCSIS 2.0 のサービス フローに関するログ テーブルを維持しなければなりません。このテーブルのエントリは、期限切れになるか、またはテーブルが満杯になるまでそのまま残ります。テーブルが満杯になった場合は、新しいエントリのためのスペースを確保するために、一番古いエントリが削除されます。

次の手順で、削除されたサービス フローのログをイネーブルに設定し、ログ テーブル (DOCS-QOS-MIB の docsQosServiceFlowLogTable) のエントリを表示します。

ステップ 1 Cisco CMTS ルータのコンソールでは、グローバル コンフィギュレーション プロンプトに `cable sflog` コマンドを入力することによって、削除されたサービス フローのログをイネーブルにします。このコマンドの構文は、次のとおりです。

```
cable sflog max-entry number entry-duration time
```

次のパラメータを指定する必要があります。

- **max-entry number** — サービス フロー ログの最大エントリ数を指定します。ログが満杯になると、新しいエントリのためのスペースを確保するために、一番古いエントリが削除されます。有効な範囲は 0 ~ 59999 で、デフォルトは 0 (サービス フロー ログをディセーブル) です。



(注) **max-entry** の値は、Cisco uBR7100 シリーズおよび Cisco uBR7200 シリーズ ルータのシャーシ全体に適用されます。Cisco uBR10012 ルータでは個々のケーブル ラインカードに適用されます。

- **entry-duration time** — サービス フロー ログにエントリを残しておく時間を秒数で指定します。CMTS はこの値より古いエントリをログから削除します。有効な範囲は 1 ~ 86400 秒です。デフォルトは 3600 秒 (1 時間) です。

次に、20,000 エントリ分のスペースがあるテーブルを使用して削除されたサービス フローのログをイネーブルに設定し、さらにテーブル内で 2 時間経過したエントリを自動的に削除する例を示します。

```
Router(config)# cable sflog max-entry 20000 entry-duration 7200
```

ステップ 2 ネットワーク管理ステーションは一定間隔で `docsQosServiceFlowLogTable` のポーリングを実行し、削除されたサービス フローの情報を収集する必要があります。テーブル エントリには固有の 32 ビット インデックスが与えられ、最大値に到達すると 0 に戻ります。

次に、2 つの削除されたサービス フローに対して 2 つのエントリのある、`docsQosServiceFlowLogTable` の例を示します。

```
csh% getmany -v2c 10.17.16.1 public docsQosServiceFlowLogTable
docsQosServiceFlowLogIfIndex.180001 = 10
docsQosServiceFlowLogIfIndex.180002 = 10
docsQosServiceFlowLogSFID.180001 = 3
docsQosServiceFlowLogSFID.180002 = 4
docsQosServiceFlowLogCmMac.180001 = 00 00 39 42 b2 56
docsQosServiceFlowLogCmMac.180002 = 00 00 39 42 b2 56
docsQosServiceFlowLogPkts.180001 = 0
docsQosServiceFlowLogPkts.180002 = 0
docsQosServiceFlowLogOctets.180001 = 0
docsQosServiceFlowLogOctets.180002 = 0
docsQosServiceFlowLogTimeDeleted.180001 = 58800
docsQosServiceFlowLogTimeDeleted.180002 = 58800
docsQosServiceFlowLogTimeCreated.180001 = 9400
docsQosServiceFlowLogTimeCreated.180002 = 9400
docsQosServiceFlowLogTimeActive.180001 = 474
docsQosServiceFlowLogTimeActive.180002 = 474
docsQosServiceFlowLogDirection.180001 = upstream(2)
docsQosServiceFlowLogDirection.180002 = downstream(1)
docsQosServiceFlowLogPrimary.180001 = true(1)
docsQosServiceFlowLogPrimary.180002 = true(1)
docsQosServiceFlowLogServiceClassName.180001 =
docsQosServiceFlowLogServiceClassName.180002 =
docsQosServiceFlowLogPolicedDropPkts.180001 = 0
```

```
docsQosServiceFlowLogPolicedDropPkts.180002 = 0
docsQosServiceFlowLogPolicedDelayPkts.180001 = 8
docsQosServiceFlowLogPolicedDelayPkts.180002 = 0
docsQosServiceFlowLogControl.180001 = active(1)
docsQosServiceFlowLogControl.180002 = active(1)
```

```
csh%
```

スペクトル管理の監視

Cisco IOS Release 12.2(8)BC2 以降のリリースを使用している場合、SNMP を使用して CISCO-CABLE-SPECTRUM-MIB にアクセスし、Cisco uBR-MC16S など、ハードウェアベースのスペクトルアナライザが組み込まれているケーブルインターフェイスカード上のスペクトル管理動作をモニタできます。MIB を使用すると、次の作業を実行できます。

- [スペクトル管理のイネーブル化 \(p.A-24\)](#)
- [スペクトル要求の結果表示 \(p.A-27\)](#)
- [個々の CM における CNR 測定結果の監視 \(p.A-28\)](#)
- [周波数ホップ情報の表示 \(p.A-30\)](#)

スペクトル管理のイネーブル化

Cisco uBR-MC16C、Cisco uBR-MC16U/X、Cisco uBR-MC28U/X、または Cisco uBR-MC5X20S/U ケーブルインターフェイスカード上でアップストリームのスペクトル管理をイネーブルにするには、ccsSpectrumRequestTable にアップストリーム用のエントリを作成します。表 A-5 に、ccsSpectrumRequestTable テーブルの各エントリに対して設定できるオブジェクトを示します。

表 A-5 ccsSpectrumRequestTable の属性

属性	タイプ	説明
ccsSpectrumRequestIndex	Integer32	各テーブル エントリを固有のものとして識別するための任意のインデックス
ccsSpectrumRequestIfIndex	InterfaceIndexOrZero	ハードウェア スペクトル管理をサポートするケーブルインターフェイスラインカード上のアップストリームを識別するための IfIndex
ccsSpectrumRequestMacAddr	MacAddress	特定の CM の Signal-to-Noise Ratio (SNR; 信号対雑音比)または Carrier-to-Noise Ratio (CNR; 搬送波対雑音比)を要求する場合は MAC アドレス、またはスペクトル全体の背景雑音を要求する場合は 0000.0000.0000
ccsSpectrumRequestUpperFreq	CCSFrequency	監視する周波数範囲の上の周波数 (KHz) (5000 ~ 42000 KHz、デフォルトは 42000 KHz)
ccsSpectrumRequestLowFreq	CCSFrequency	監視する周波数範囲の下の周波数 (KHz) (5000 ~ 42000 KHz、デフォルトは 5000 KHz)
ccsSpectrumRequestResolution	Integer32	周波数範囲のサンプリング方法を決定する、要求された分解能 (12 ~ 37000 KHz、デフォルトは 60 KHz)
ccsSpectrumRequestStartTime	TimeStamp	スペクトル測定開始時刻
ccsSpectrumRequestStoppedTime	TimeStamp	スペクトル測定終了時刻
ccsSpectrumRequestOperation	CCSRequestOperation	新しいスペクトル管理要求の開始または現在の要求の中止

表 A-5 ccsSpectrumRequestTable の属性 (続き)

属性	タイプ	説明
ccsSpectrumRequestOperState	CCSRequestOperState	現在のスペクトル管理要求の動作状態を提供
ccsSpectrumRequestStatus	RowStatus	テーブル エントリの変更、作成、および削除を制御

アップストリームのスペクトル管理をイネーブルにする手順は、次のとおりです。

- ステップ 1** 所定のアップストリームの ifIndex を取得します。それには ifDescr に対して GET 要求を行います。次に、Cisco uBR7246VXR ルータの出力例を示します。このルータにはファストイーサネットポートアダプタが1つ、スロット6にCisco uBR-MC16S ケーブルインターフェイスカードが1つ搭載されています。

```

csh% getmany -v2c 10.10.10.13 public ifDescr

ifDescr.1 = FastEthernet0/0
ifDescr.2 = FastEthernet0/1
ifDescr.3 = FastEthernet1/0
ifDescr.4 = FastEthernet1/1
ifDescr.5 = Cable6/0
ifDescr.6 = Cable6/0-upstream0
ifDescr.7 = Cable6/0-upstream1
ifDescr.8 = Cable6/0-upstream2
ifDescr.9 = Cable6/0-upstream3
ifDescr.10 = Cable6/0-upstream4
ifDescr.11 = Cable6/0-upstream5

csh%

```

- ステップ 2** 所定のアップストリームに対応する行を ccsSpectrumRequestTable に作成します。最小限、待機条件の行を作成し、その行をアクティブにする前にアップストリームの ifIndex を設定する必要があります。

次に、アップストリーム 2 (ifIndex 8) 用の行を作成する例を示します。この行には、任意の行インデックスとして 8 が選択されています。必ず、未使用の行インデックスを選択してください。

```

csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestStatus.8 -i 5
ccsSpectrumRequestStatus.8 = wait(5)

csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestIfIndex.8 -i 8
ccsSpectrumRequestIfIndex.8 = 8

csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestStatus.8 -i 1
ccsSpectrumRequestStatus.8 = active(1)

csh%

```



- (注)** ハードウェア スペクトル管理をサポートするケーブル インターフェイス ラインカード上の、アップストリームを指定しない IfIndex を使用しようとする、無効値エラーによって SET 要求が失敗します。

ステップ 3 新しい行エントリの現在の設定を表示し、デフォルトパラメータが妥当かどうかを確認します。

```
csh% getmany -v2c 10.10.10.13 public ccsSpectrumRequestTable

ccsSpectrumRequestIfIndex.8 = 8
ccsSpectrumRequestMacAddr.8 = 00 00 00 00 00 00
ccsSpectrumRequestLowFreq.8 = 5000
ccsSpectrumRequestUpperFreq.8 = 42000
ccsSpectrumRequestResolution.8 = 60
ccsSpectrumRequestOperation.8 = none(0)
ccsSpectrumRequestOperState.8 = idle(0)
ccsSpectrumRequestStartTime.8 = 0
ccsSpectrumRequestStoppedTime.8 = 0
ccsSpectrumRequestStatus.8 = active(1)

csh%
```

ステップ 4 `ccsSpectrumRequestMacAddr` はデフォルトで、すべてゼロに設定されます。この場合、アップストリーム全体に背景雑音が要求されます。特定の CM で CNR をモニタする場合は、`ccsSpectrumRequestMacAddr` をその CM の MAC アドレスに設定します。

```
csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestMacAddr.8 -o '00 01 64 ff eb 95'
ccsSpectrumRequestMacAddr.3 = 00 01 64 ff eb 95

csh% getmany -v2c 10.10.10.13 public ccsSpectrumRequestTable

ccsSpectrumRequestIfIndex.8 = 8
ccsSpectrumRequestMacAddr.8 = 00 01 64 ff eb 95
ccsSpectrumRequestLowFreq.8 = 5000
ccsSpectrumRequestUpperFreq.8 = 42000
ccsSpectrumRequestResolution.8 = 60
ccsSpectrumRequestOperation.8 = none(0)
ccsSpectrumRequestOperState.8 = idle(0)
ccsSpectrumRequestStartTime.8 = 0
ccsSpectrumRequestStoppedTime.8 = 0
ccsSpectrumRequestStatus.8 = active(1)

csh%
```

ステップ 5 他のパラメータをデフォルト値から変更しなければならない場合は、適切な値に設定します。次に、周波数範囲をデフォルトの範囲（5～42 MHz）から 20～28 MHz に変更する例を示します。

```
csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestLowFreq.8 -i 20000
ccsSpectrumRequestLowFreq.8 = 20000

csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestUpperFreq.8 -i 28000
ccsSpectrumRequestUpperFreq.8 = 28000

csh%
```

ステップ 6 すべてのパラメータが正しい場合は、`ccsSpectrumRequestOperation` オブジェクトを `start(1)` に設定し、アップストリームのスペクトルモニタを開始します。

```
csh% setany -v2c 10.10.10.13 private ccsSpectrumRequestOperation.8 -i 1
ccsSpectrumRequestOperation.8 = start(1)

csh%
```

スペクトル要求の結果表示

スペクトル要求の結果を表示するには、`ccsSpectrumDataTable` の対応する行のオブジェクトを表示します (`show controllers cable upstream spectrum` コマンドの出力と同じ情報が表示されます)。表 A-6 に、このテーブルに格納されているオブジェクトを示します。

表 A-6 `ccsSpectrumDataTable` の属性

属性	タイプ	説明
<code>ccsSpectrumDataFreq</code>	<code>CCSMeasuredFrequency</code>	この電力測定が行われた周波数 (KHz)
<code>ccsSpectrumDataPower</code>	<code>INTEGER</code>	所定の周波数で測定された受信電力 (-50 ~ 50 dBmV)

スペクトル要求の結果を表示する手順は、次のとおりです。

ステップ 1 「スペクトル管理のイネーブル化」 (p.A-24) で説明したように、`ccsSpectrumRequestTable` に行を追加することによって、スペクトル要求を作成してアクティブにします。

ステップ 2 `ccsSpectrumDataTable` 内のエントリを表示します。

```
csh% getmany -v2c 10.10.10.13 public ccsSpectrumDataTable

ccsSpectrumDataFreq.8.20001 = 20001
ccsSpectrumDataFreq.8.20121 = 20121
ccsSpectrumDataFreq.8.20241 = 20241
ccsSpectrumDataFreq.8.20361 = 20361

...

ccsSpectrumDataFreq.8.27561 = 27561
ccsSpectrumDataFreq.8.27681 = 27681
ccsSpectrumDataFreq.8.27801 = 27801
ccsSpectrumDataFreq.8.27921 = 27921

ccsSpectrumDataPower.8.20001 = -43
ccsSpectrumDataPower.8.20121 = -50
ccsSpectrumDataPower.8.20241 = -47
ccsSpectrumDataPower.8.20361 = -46

...

ccsSpectrumDataPower.8.27561 = -47
ccsSpectrumDataPower.8.27681 = -44
ccsSpectrumDataPower.8.27801 = -46
ccsSpectrumDataPower.8.27921 = -42

csh%
```



(注) `ccsSpectrumDataTable` のエントリは、`ccsSpectrumRequestTable` 内のスペクトル要求エントリに対応する行番号および電力測定が行われた周波数 (KHz) がインデックスになります。

ステップ 3 `ccsSpectrumDataFreq` の値を使用して、電力測定が行われた周波数を調べます。次にこの周波数の値を `ccsSpectrumRequestTable` の行エントリとともに使用して、所定の周波数に固有の電力測定値を調べます。

たとえば、次の行は、ccsSpectrumRequestTable の行 3 で指定されたアップストリームの電力測定結果が 27.801 MHz で -46 dBmVであることを示しています。

```
ccsSpectrumDataPower.8.27801 = -46
```

個々の CM における CNR 測定結果の監視

個々の CM における CNR を調べるには、ccsSNRRequestTable にエントリを作成してアクティブにします。表 A-7 に、このテーブルの各エントリに設定できるオブジェクトを示します。

表 A-7 ccsSNRRequestTable の属性

属性	タイプ	説明
ccsSNRRequestIndex	Integer32	各テーブルエントリを固有のものとして識別するための任意のインデックス
ccsSNRRequestMacAddr	MacAddress	レポート対象となるオンラインのリモート CM の MAC アドレス
ccsSNRRequestSNR	Integer32	測定された SNR 値 (dB)。動作ステートが [running] の場合、この値は 0 です。
ccsSNRRequestOperation	CCSRequestOperation	現在の動作を設定 : start (開始)、pending (保留)、running (実行中)、または abort (中止)
ccsSNRRequestOperState	CCSRequestOperState	現在の動作ステートを報告 : idle、pending、running、noError、aborted、notOnLine、invalidMac、timeOut、fftBusy、fftFailed、others
ccsSNRRequestStartTime	TimeStamp	SNR 測定動作を開始した時刻
ccsSNRRequestStoppedTime	TimeStamp	SNR 測定動作を停止した時刻
ccsSNRRequestStatus	RowStatus	テーブルエントリの変更、作成、および削除を制御

特定の CM における CNR 情報を調べる手順は、次のとおりです。

- ステップ 1** 所定の CM に対応する行を ccsSNRRequestTable に作成します。最小限、待機条件の行を作成し、その行をアクティブにする前に CM の MAC アドレスを設定する必要があります。次に例を示します。

```

csh% setany -v2c 10.10.10.13 private ccsSNRRequestStatus.200 -i 5
ccsSNRRequestStatus.200 = createAndWait(5)

csh% setany -v2c 10.10.10.13 private ccsSNRRequestMacAddr.200 -o '00 03 e3 50 9b 3d'
ccsSNRRequestMacAddr.200 = 00 03 e3 50 9b 3d

csh% setany -v2c 10.10.10.13 private ccsSNRRequestStatus.200 -i 1
ccsSNRRequestStatus.200 = active(1)

csh%
```

ステップ 2 新しい行エントリの現在の設定を表示し、デフォルトパラメータが妥当かどうかを確認します。

```
csh% getmany -v2c 10.10.10.13 public ccsSNRRequestTable

ccsSNRRequestMacAddr.200 = 00 03 e3 50 9b 3d
ccsSNRRequestSNR.200 = 0
ccsSNRRequestOperation.200 = none(0)
ccsSNRRequestOperState.200 = idle(0)
ccsSNRRequestStartTime.200 = 0
ccsSNRRequestStoppedTime.200 = 0
ccsSNRRequestStatus.200 = active(1)

csh%
```

ステップ 3 すべてのパラメータが正しい場合は、ccsSpectrum ccsSNRRequestOperation オブジェクトを start(1) に設定し、CM の監視を開始します。

```
csh% setany -v2c 10.10.10.13 private ccsSNRRequestOperation.200 -i 1
ccsSNRRequestOperation.200 = start(1)

csh%
```

ステップ 4 ccsSNRRequestOperState が noError を示すまで、ccsSNRRequestTable のポーリングを繰り返します。noError の時点で、ccsSNRRequestSNR は CM の現在の CNR 値を示します。



(注) ccsSNRRequestSNR オブジェクトは、ccsSNRRequestOperState のステータスが running(2) であるかぎり 0 を示します。

```
csh% getmany -v2c 10.10.10.13 public ccsSNRRequestTable

ccsSNRRequestMacAddr.200 = 00 03 e3 50 9b 3d
ccsSNRRequestSNR.200 = 0
ccsSNRRequestOperation.200 = start(1)
ccsSNRRequestOperState.200 = running(2)
ccsSNRRequestStartTime.200 = 0
ccsSNRRequestStoppedTime.200 = 0
ccsSNRRequestStatus.200 = active(1)

csh% getmany -v2c 10.10.10.13 public ccsSNRRequestTable

ccsSNRRequestMacAddr.200 = 00 03 e3 50 9b 3d
ccsSNRRequestSNR.200 = 25
ccsSNRRequestOperation.200 = start(1)
ccsSNRRequestOperState.200 = noError(3)
ccsSNRRequestStartTime.200 = 298853
ccsSNRRequestStoppedTime.200 = 298974
ccsSNRRequestStatus.200 = active(1)

csh%
```

周波数ホップ情報の表示

アップストリームの最新周波数ホップ情報を調べるには、ccsUpSpecMgmtTable のオブジェクトを表示します。表 A-8 に、このテーブルの属性を示します。

表 A-8 ccsUpSpecMgmtEntry の属性

属性	タイプ	説明
ccsUpSpecMgmtHopPriority	INTEGER	アップストリームの雑音が多すぎる場合の修正方法を決定する、周波数、変調プロファイル、およびチャンネル幅のプライオリティを指定 (デフォルトでは周波数、変調プロファイル、チャンネル幅の順)
ccsUpSpecMgmtSnrThres1	Integer32	変調プロファイル 1 の上限 SNR しきい値を指定 (5 ~ 35 dB、デフォルトは 25)
ccsUpSpecMgmtSnrThres2	Integer32	変調プロファイル 2 の上限 SNR しきい値を指定 (5 ~ 35 dB、デフォルトは 15、ccsUpSpecMgmtSnrThres1 で指定した値より小さくすることが必要)
ccsUpSpecMgmtFecCorrectThres1	Integer32	変調プロファイル 1 の Forward Error Correction (FEC; 前方エラー訂正) 訂正可能エラーのしきい値を指定 (1 ~ 20%)
ccsUpSpecMgmtFecCorrectThres2	Integer32	廃止、使用不可
ccsUpSpecMgmtFecUnCorrectThres1	Integer32	変調プロファイル 1 の FEC 訂正不能エラーのしきい値を指定 (1 ~ 20%)
ccsUpSpecMgmtFecUnCorrectThres2	Integer32	廃止、使用不可
ccsUpSpecMgmtSnrPollPeriod	Integer32	廃止、使用不可
ccsUpSpecMgmtHopCondition ¹	INTEGER	周波数ホップを引き起こす条件を報告 (SNR 値またはオフラインになったモデムの割合)
ccsUpSpecMgmtFromCenterFreq ¹	CCSFrequency	直前の周波数ホップ前の中心周波数 (KHz)
ccsUpSpecMgmtToCenterFreq ¹	CCSFrequency	直前の周波数ホップ後の現在の中心周波数 (KHz)
ccsUpSpecMgmtFromBandWidth ¹	CCSFrequency	直前の周波数ホップ前のチャンネル幅 (KHz)
ccsUpSpecMgmtToBandWidth ¹	CCSFrequency	直前の周波数ホップ後の現在のチャンネル幅 (KHz)
ccsUpSpecMgmtFromModProfile ¹	Integer32	直前の周波数ホップ前の変調プロファイル番号
ccsUpSpecMgmtToModProfile ¹	Integer32	直前の周波数ホップ後の現在の変調プロファイル番号
ccsUpSpecMgmtSNR	Integer32	アップストリームの現在の SNR 値 (dB)

1. アップストリームで周波数ホップが発生したときに送信される通知メッセージで、これらのオブジェクトも送信されます。

1 つまたは複数のアップストリームについて周波数ホップデータを収集する手順は、次のとおりです。

- ステップ 1** 所定のアップストリームの ifIndex を取得します。それには ifDescr に対して GET 要求を行います。次に、Cisco uBR7246VXR ルータのスロット 5/0 に搭載されたケーブル インターフェイス カードの最初の 4 つのアップストリームに関する出力例を示します。

```
csch% getmany -v2c 10.10.10.13 public ifDescr
...
ifDescr.24 = Cable5/0-upstream0
ifDescr.25 = Cable5/0-upstream1
ifDescr.26 = Cable5/0-upstream2
ifDescr.27 = Cable5/0-upstream3
...
csch%
```

- ステップ 2** ccsUpSpecMgmtTable を表示します。所定のアップストリーム（複数可）の ifDescr 値を使用して、アップストリームの値を調べます。次に、上の 4 つのアップストリームに関連する出力例を示します。

```
csch% getmany -v2c 10.10.10.13 public ccsUpSpecMgmtTable
...
ccsUpSpecMgmtHopPriority.24 = frqModChannel(0)
ccsUpSpecMgmtHopPriority.25 = frqModChannel(0)
ccsUpSpecMgmtHopPriority.26 = frqModChannel(0)
ccsUpSpecMgmtHopPriority.27 = frqModChannel(0)
...
ccsUpSpecMgmtSnrThres1.24 = 25
ccsUpSpecMgmtSnrThres1.25 = 25
ccsUpSpecMgmtSnrThres1.26 = 25
ccsUpSpecMgmtSnrThres1.27 = 25
...
ccsUpSpecMgmtSnrThres2.24 = 15
ccsUpSpecMgmtSnrThres2.25 = 15
ccsUpSpecMgmtSnrThres2.26 = 15
ccsUpSpecMgmtSnrThres2.27 = 15
...
ccsUpSpecMgmtFecCorrectThres1.24 = 1
ccsUpSpecMgmtFecCorrectThres1.25 = 1
ccsUpSpecMgmtFecCorrectThres1.26 = 1
ccsUpSpecMgmtFecCorrectThres1.27 = 1
...
ccsUpSpecMgmtFecCorrectThres2.24 = 1
ccsUpSpecMgmtFecCorrectThres2.25 = 1
ccsUpSpecMgmtFecCorrectThres2.26 = 1
ccsUpSpecMgmtFecCorrectThres2.27 = 1
...
ccsUpSpecMgmtFecUnCorrectThres1.24 = 1
ccsUpSpecMgmtFecUnCorrectThres1.25 = 1
ccsUpSpecMgmtFecUnCorrectThres1.26 = 1
ccsUpSpecMgmtFecUnCorrectThres1.27 = 1
...
ccsUpSpecMgmtFecUnCorrectThres2.24 = 1
ccsUpSpecMgmtFecUnCorrectThres2.25 = 1
ccsUpSpecMgmtFecUnCorrectThres2.26 = 1
ccsUpSpecMgmtFecUnCorrectThres2.27 = 1
...
ccsUpSpecMgmtSnrPollPeriod.24 = 15
ccsUpSpecMgmtSnrPollPeriod.25 = 15
ccsUpSpecMgmtSnrPollPeriod.26 = 15
ccsUpSpecMgmtSnrPollPeriod.27 = 15
...
```

```
ccsUpSpecMgmtHopCondition.24 = snr(0)
ccsUpSpecMgmtHopCondition.25 = snr(0)
ccsUpSpecMgmtHopCondition.26 = snr(0)
ccsUpSpecMgmtHopCondition.27 = snr(0)
...
ccsUpSpecMgmtFromCenterFreq.24 = 10000
ccsUpSpecMgmtFromCenterFreq.25 = 15008
ccsUpSpecMgmtFromCenterFreq.26 = 20000
ccsUpSpecMgmtFromCenterFreq.27 = 25008
...
ccsUpSpecMgmtToCenterFreq.24 = 10000
ccsUpSpecMgmtToCenterFreq.25 = 15008
ccsUpSpecMgmtToCenterFreq.26 = 20000
ccsUpSpecMgmtToCenterFreq.27 = 25008
...
ccsUpSpecMgmtFromBandWidth.24 = 1600
ccsUpSpecMgmtFromBandWidth.25 = 3200
ccsUpSpecMgmtFromBandWidth.26 = 3200
ccsUpSpecMgmtFromBandWidth.27 = 3200
...
ccsUpSpecMgmtToBandWidth.24 = 1600
ccsUpSpecMgmtToBandWidth.25 = 3200
ccsUpSpecMgmtToBandWidth.26 = 3200
ccsUpSpecMgmtToBandWidth.27 = 3200
...
ccsUpSpecMgmtFromModProfile.24 = 1
ccsUpSpecMgmtFromModProfile.25 = 1
ccsUpSpecMgmtFromModProfile.26 = 1
ccsUpSpecMgmtFromModProfile.27 = 1
...
ccsUpSpecMgmtToModProfile.24 = 2
ccsUpSpecMgmtToModProfile.25 = 2
ccsUpSpecMgmtToModProfile.26 = 2
ccsUpSpecMgmtToModProfile.27 = 2
...
ccsUpSpecMgmtSNR.24 = 0
ccsUpSpecMgmtSNR.25 = 0
ccsUpSpecMgmtSNR.26 = 0
ccsUpSpecMgmtSNR.27 = 0

csh%
```

フラップリストの使用法

Cisco CMTS ルータ上でフラップリストの設定、消去、アクセスを行う手順は、次のとおりです。

- [SNMP によるフラップリストの動作設定 \(p.A-33\)](#)
- [SNMP によるフラップリストの表示 \(p.A-33\)](#)
- [特定の CM におけるフラップリスト情報の表示 \(p.A-35\)](#)
- [SNMP によるフラップリストおよびカウンタの消去 \(p.A-37\)](#)

SNMP によるフラップリストの動作設定

SNMP を使用して Cisco CMTS 上でフラップリストトラブルシューティング機能を設定するには、CISCO-CABLE-SPECTRUM-MIB の適切な `cssFlapObjects` 属性を設定します。表 A-9 に、設定できる属性を示します。

表 A-9 フラップリストを設定するための属性

属性	タイプ	範囲	説明
<code>cssFlapListMaxSize</code>	Integer32	1 ~ 65536 ¹	フラップリストがサポートできるモデムの最大数。デフォルトは 100 です。
<code>cssFlapListCurrentSize</code>	Integer32	1 ~ 65536 ¹	フラップリストに現在含まれているモデムの数
<code>cssFlapAging</code>	Integer32	1 ~ 86400	フラップエントリのエージングしきい値 (分)。デフォルトは 10080 分 (180 時間すなわち 7 日) です。
<code>cssFlapInsertionTime</code>	Integer32	60 ~ 86400	最悪の場合のインサート時間 (秒)。CM がこの時間内に登録ステージを完了しなかった場合、その CM はフラップリストに追加されます。デフォルト値は 90 秒です。
<code>cssFlapPowerAdjustThreshold</code>	Integer32	1 ~ 10	モデムの出力が出力調整しきい値を超えて調整されると、そのモデムはフラップリストに追加されます。
<code>cssFlapMissThreshold</code>	Unsigned32	1 ~ 12	CM が連続してこの回数だけ MAC レイヤステーションメンテナンス (キープアライブ) メッセージを確認しなかった場合、その CM はフラップリストに追加されます。

1. SNMP を使用する場合、これらのパラメータに指定できる範囲は 1 ~ 65536 (32 ビット値) ですが、有効な動作範囲は 1 ~ 8191 です。

SNMP によるフラップリストの表示

SNMP を使用してフラップリストの内容を表示するには、CISCO-CABLE-SPECTRUM-MIB の `cssFlapTable` を調べます。このテーブルには、CM ごとにエントリが 1 つずつあります。表 A-10 で、このテーブルの各属性について簡単に説明します。

■ フラップリストの使用方法

表 A-10 cssFlapTable の属性

属性	タイプ	説明
cssFlapMacAddr	MacAddress	CM のケーブル インターフェイスの MAC アドレス。フラッピング CM に対応するフラップリスト エントリを特定します。
ccsFlapUpstreamIfIndex	InterfaceIndex	フラッピング CM が使用しているアップストリーム
ccsFlapDownstreamIfIndex	InterfaceIndex	フラッピング CM が使用しているダウンストリーム
ccsFlapLastFlapTime	DateAndTime	CM のフラッピングが最後に発生した時刻を示すタイムスタンプ
ccsFlapCreateTime	DateAndTime	このエントリがテーブルに追加されたときのタイムスタンプ
ccsFlapRowStatus	RowStatus	このエントリのステータスに関連する属性を制御
ccsFlapInsertionFailNum	Unsigned32	CM がアクティブになってネットワークに参加した回数。このカウンタは、最初のリンク確立から再確立までの時間が、 cable flap-list insertion-time コマンドまたは ccsFlapInsertionTime 属性を使用して設定されたしきい値 パラメータを下回ったときに増分します。 CM がインサート時間 (ccsFlapInsertionTime) 内に登録を完了できなかった場合は、初期メンテナンス パケットが再送信されます。CMTS は予測より早くパケットを受信すると、このカウンタを増分します。
ccsFlapHitNum	Unsigned32	CM が MAC レイヤ ステーション メンテナンス (キープアライブ) メッセージに回答した回数 (最小ヒット レートは 30 秒に 1 回)
ccsFlapMissNum	Unsigned32	CM が MAC レイヤ ステーション メンテナンス (キープアライブ) メッセージに回答しなかった回数。Cisco ケーブル インターフェイス ラインカードの場合、8% のミス レートは正常です。CMTS が 25 ミリ秒以内にレンジング要求をミスすると、ミス回数が増えます。
ccsFlapCrcErrorNum	Unsigned32	CMTS アップストリーム レシーバがパケットに CRC エラーのフラグを設定した回数。この値が大きい場合、ケーブルのアップストリームで雑音レベルが高くなっている可能性があります。モデムがまだフラッピングしていなくても、問題になる可能性があります。
ccsFlapPowerAdjustmentNum	Unsigned32	ステーション メンテナンス中に CM のアップストリーム送信出力が調整された回数。調整が出力調整しきい値を超えると、この値が増えます。
ccsFlapTotalNum	Unsigned32	モデムのフラッピング回数 (下記の合計) <ul style="list-style-type: none"> • ccsFlapInsertionFailNum が増加した回数 • CMTS がミスに続いてヒットを受信した回数 • ccsFlapPowerAdjustmentNum が増加した回数
ccsFlapResetNow	Boolean	このオブジェクトを True(1) に設定すると、すべてのフラップ リストカウンタがゼロにリセットされます。
ccsFlapLastResetTime	DateAndTime	このエントリに対応するすべてのカウンタがゼロにリセットされたときのタイムスタンプ

次に、MAC アドレスが 00.07.0E.02.CA.91 (0.7.14.2.202.145) および 00.07.0E.03.68.89 (0.7.14.3.104.137) という 2 つの CM に対応するエントリが含まれている、ccsFlapTable の出力例を示します。

```
csh% getmany -v2c 10.10.11.12 public ccsFlapTable

ccsFlapUpstreamIfIndex.0.7.14.2.202.145 = 17
ccsFlapUpstreamIfIndex.0.7.14.3.104.137 = 17
ccsFlapDownstreamIfIndex.0.7.14.2.202.145 = 21
ccsFlapDownstreamIfIndex.0.7.14.3.104.137 = 21
ccsFlapInsertionFails.0.7.14.2.202.145 = 2
ccsFlapInsertionFails.0.7.14.3.104.137 = 0
ccsFlapHits.0.7.14.2.202.145 = 54098
ccsFlapHits.0.7.14.3.104.137 = 54196
ccsFlapMisses.0.7.14.2.202.145 = 65
ccsFlapMisses.0.7.14.3.104.137 = 51
ccsFlapCrcErrors.0.7.14.2.202.145 = 0
ccsFlapCrcErrors.0.7.14.3.104.137 = 0
ccsFlapPowerAdjustments.0.7.14.2.202.145 = 0
ccsFlapPowerAdjustments.0.7.14.3.104.137 = 0
ccsFlapTotal.0.7.14.2.202.145 = 5
ccsFlapTotal.0.7.14.3.104.137 = 4
ccsFlapLastFlapTime.0.7.14.2.202.145 = 14 03 04 1e 07 35 10 00
ccsFlapLastFlapTime.0.7.14.3.104.137 = 14 03 04 1e 07 34 12 00
ccsFlapCreateTime.0.7.14.2.202.145 = 14 03 04 1e 07 00 2b 00
ccsFlapCreateTime.0.7.14.3.104.137 = 14 03 04 1e 07 00 2c 00
ccsFlapRowStatus.0.7.14.2.202.145 = 1
ccsFlapRowStatus.0.7.14.3.104.137 = 1
ccsFlapInsertionFailNum.0.7.14.2.202.145 = 2
ccsFlapInsertionFailNum.0.7.14.3.104.137 = 0
ccsFlapHitNum.0.7.14.2.202.145 = 54098
ccsFlapHitNum.0.7.14.3.104.137 = 54196
ccsFlapMissNum.0.7.14.2.202.145 = 65
ccsFlapMissNum.0.7.14.3.104.137 = 51
ccsFlapCrcErrorNum.0.7.14.2.202.145 = 0
ccsFlapCrcErrorNum.0.7.14.3.104.137 = 0
ccsFlapPowerAdjustmentNum.0.7.14.2.202.145 = 0
ccsFlapPowerAdjustmentNum.0.7.14.3.104.137 = 0
ccsFlapTotalNum.0.7.14.2.202.145 = 5
ccsFlapTotalNum.0.7.14.3.104.137 = 4
ccsFlapResetNow.0.7.14.2.202.145 = 2
ccsFlapResetNow.0.7.14.3.104.137 = 2
ccsFlapLastResetTime.0.7.14.2.202.145 = 14 03 04 1e 06 39 0c 00
ccsFlapLastResetTime.0.7.14.3.104.137 = 14 03 04 1e 06 39 1e 00

csh%
```



ヒント

フラップリストのコンフィギュレーションパラメータとフラップリストの内容を両方とも収集するには、ccsFlapObjects に対する GET 要求を実行します。

特定の CM におけるフラップリスト情報の表示

SNMP 要求を使用して特定の CM に関するフラップリスト情報を表示するには、ccsFlapTable からエントリを取り出すためのインデックスとして、CM の MAC アドレスを使用します。特定の CM に対応するフラップリストエントリを取り出す手順は、次のとおりです。

詳細な手順

- ステップ 1** CM の MAC アドレスをドット付き 10 進表記に変換します。たとえば、MAC アドレス 000C.64ff.eb95 の場合は 0.12.100.255.235.149 になります。

■ フラップリストの使用法

ステップ 2 `ccsFlapTable` の情報を要求するためのインスタンスとして、ドット付き 10 進形式の MAC アドレスを使用します。たとえば、この CM に対応する `ccsFlapHits`、`ccsFlapMisses`、および `ccsFlapPowerAdjustments` の値を取り出すには、次のオブジェクトに対する SNMP 要求を行います。

- `ccsFlapHits.0.12.100.255.235.149`
- `ccsFlapMisses.0.12.100.255.235.149`
- `ccsFlapPowerAdjustments.0.12.100.255.235.149`

例

MAC アドレス `000C.64ff.eb95` に対する `show cable flap-list` コマンドと同じフラップリスト情報を取り出すものとします。

```
Router# show cable flap-list
```

```
MAC Address      Upstream      Ins   Hit   Miss  CRC   P-Adj Flap  Time
000C.64ff.eb95  Cable3/0/U4  3314  55605 50460 0      *42175 47533 Jan 27 02:49:10
```

```
Router#
```

SNMP ツールを使用して、`ccsFlapTable` を取り出し、10 進形式の MAC アドレスでフィルタリングします。次に、標準 UNIX の `getone` コマンドを使用する場合の入力例を示します。

```
csh% getmany -v2c 192.168.100.121 public ccsFlapTable | grep 0.12.100.255.235.149
```

```
ccsFlapUpstreamIfIndex.0.12.100.255.235.149 = 15
ccsFlapDownstreamIfIndex.0.12.100.255.235.149 = 17
ccsFlapInsertionFails.0.12.100.255.235.149 = 3315
ccsFlapHits.0.12.100.255.235.149 = 55608
ccsFlapMisses.0.12.100.255.235.149 = 50460
ccsFlapCrcErrors.0.12.100.255.235.149 = 0
ccsFlapPowerAdjustments.0.12.100.255.235.149 = 42175
ccsFlapTotal.0.12.100.255.235.149 = 47534
ccsFlapLastFlapTime.0.12.100.255.235.149 = 07 d4 01 1b 02 33 1a 00
ccsFlapCreateTime.0.12.100.255.235.149 = 07 d4 01 16 03 23 22 00
ccsFlapRowStatus.0.12.100.255.235.149 = active(1)
ccsFlapInsertionFailNum.0.12.100.255.235.149 = 3315
ccsFlapHitNum.0.12.100.255.235.149 = 55608
ccsFlapMissNum.0.12.100.255.235.149 = 50460
ccsFlapCrcErrorNum.0.12.100.255.235.149 = 0
ccsFlapPowerAdjustmentNum.0.12.100.255.235.149 = 42175
ccsFlapTotalNum.0.12.100.255.235.149 = 47534
ccsFlapResetNow.0.12.100.255.235.149 = false(2)
ccsFlapLastResetTime.0.12.100.255.235.149 = 07 d4 01 16 03 20 18 00
```

```
csh%
```

特定の値を 1 つだけ求める場合は、そのオブジェクトのインスタンスとして 10 進形式の MAC アドレスを使用します。

```
csh% getone -v2c 172.22.85.7 public ccsFlapMisses.0.12.100.255.235.149
```

```
ccsFlapMisses.0.12.100.255.235.149 = 50736
```

```
csh %
```

SNMP によるフラップ リストおよびカウンタの消去

フラップ リストから CM を削除する場合、1 つまたはすべてのフラップ リスト カウンタを消去する場合は、CISCO-CABLE-SPECTRUM-MIB の適切な `cssFlapObjects` 属性を設定します。表 A-11 に、SNMP カウンタを消去する属性を示します。

表 A-11 フラップ リストを消去する属性

属性	タイプ	説明
<code>cssFlapResetAll</code>	Boolean	このオブジェクトを <code>True(1)</code> に設定すると、すべてのフラップ リスト カウンタがゼロにリセットされます。
<code>cssFlapClearAll</code>	Boolean	このオブジェクトを <code>True(1)</code> に設定すると、フラップ リストからすべての CM が削除され、 <code>cssFlapTable</code> 内のすべてのエントリが破棄されます。モデムのフラッピングが続いている場合は、新しいエントリとしてフラップ リストにモデムが再び追加されます。



(注) `cssFlapLastClearTime` 属性には、`cssFlapTable` テーブルのエントリが最後に削除された日時が含まれています。

加入者トラフィック管理機能の使用法

CISCO-CABLE-QOS-MONITOR-MIB の属性を使用すると、Cisco CMTS ルータ上で加入者トラフィック管理機能を設定し、設定された帯域限度に反した CM を表示できます。次の作業に必要な手順について説明します。

- [QoS に関する強制ルールの設定および表示 \(p.A-38\)](#)
- [強制ルールに違反した加入者の表示 \(p.A-39\)](#)
- [強制ルールに違反した加入者への通知 \(p.A-40\)](#)

QoS に関する強制ルールの設定および表示

ccqmCmtsEnforceRuleTable テーブルの属性を使用して、Cisco CMTS ルータ上で実行する Quality of Service (QoS; サービス品質) 強制ルールを設定します。このテーブルに対するクエリーによって、現在定義されている強制ルールを表示することもできます。

表 A-12 に、ccqmCmtsEnforceRuleTable テーブルの各行に含まれる属性を示します。強制ルールごとに 1 つずつ行エントリ (CcqmCmtsEnforceRuleEntry) があり、固有のルール名 (ccqmCmtsEnfRuleName) がインデックスになります。

表 A-12 QoS 強制ルールに関する属性

属性	タイプ	説明
ccqmCmtsEnfRuleName	DisplayString	強制ルールの固有名 (最大 15 文字)
ccqmCmtsEnfRuleRegQoS	Unsigned32	0 に設定されていない場合、この属性は docsIfQosProfileTable テーブルのインデックスになり、このルールで登録プロファイルとして使用する QoS プロファイルが特定されます。
ccqmCmtsEnfRuleEnfQoS	Unsigned32	0 に設定されていない場合、この属性は docsIfQosProfileTable テーブルのインデックスになり、加入者が各自の帯域限度に反したときに有効になる、強制プロファイルとして使用する QoS プロファイルが特定されます。
ccqmCmtsEnfRuleMonDuration	Unsigned32	帯域限度に反しているかどうかを調べるためにユーザを監視する時間枠 (分)。有効な範囲は 10 ~ 10080 分 (7 日) です。デフォルトは 360 分 (6 時間) です。
ccqmCmtsEnfRuleSampleRate	Unsigned32	帯域限度に反していないかどうかを調べるために、Cisco CMTS が加入者の帯域利用状況をチェックする頻度 (分)。有効な範囲は 10 ~ 120 分 (2 時間) です。デフォルトは 15 分です。
ccqmCmtsEnfRulePenaltyPeriod	Unsigned32	帯域限度に反したあとで、加入者をペナルティ期間にとどめておく時間 (分)。加入者はこの期間が経過するまで、または Cisco CMTS 上でペナルティ期間が手動で削除されるまで、強制 QoS プロファイルが適用されます。有効な範囲は 1 ~ 10080 分、デフォルトは 10080 (7 日) です。
ccqmCmtsEnfRuleByteCount	Unsigned32	加入者が 1 回の監視枠の時間内に送信または受信 (実行ルールの方向による) できる最大キロバイト数。加入者がこのバイト数を超えると、過剰消費のフラグが設定され、ペナルティ期間の適用を受けることになります。有効な範囲は任意の 32 ビット整数値です。デフォルトはありません。
ccqmCmtsEnfRuleDirection	CCQMRuleDirection	この強制ルールでバイト数を監視する方向 (アップストリーム、ダウンストリーム、または双方向) を指定します。

表 A-12 QoS 強制ルールに関する属性 (続き)

属性	タイプ	説明
ccqmCmtsEnfRuleAutoEnforce	TruthValue	加入者が帯域限度に反したときに、強制 QoS プロファイルを自動的に適用するかどうかを指定します。デフォルトは False (強制 QoS プロファイルを自動的に適用しない) です。
ccqmCmtsEnfRuleRowStatus	RowStatus	この行のステータスを指定します。この強制ルール エントリがアクティブになるのは、RowStatus が active(1) に設定されている場合だけです。ただし、行パラメータのいずれかを変更する場合は、最初に RowStatus を notInService(2) に設定することによって、このルールで現在行っているユーザの監視を終了する必要があります。さらに、監視を再開する前に、行を active(1) に戻さなければなりません。

強制ルールに違反した加入者の表示

表 A-13 に、ccqmEnfRuleViolateTable テーブルの各エントリ (ccqmEnfRuleViolateEntry) に使用できる属性を示します。このテーブルは、設定された監視期間中に強制ルールに違反した加入者を示します。

表 A-13 ccqmEnfRuleViolateTable の属性

属性	タイプ	説明
ccqmEnfRuleViolateID	Unsigned32	行エントリに割り当てられた固有の ID。DOCSIS 1.0 ユーザの場合、加入者の Service Flow (SFID) と同じ値になります。
ccqmEnfRuleViolateMacAddr	MacAddress	QoS の帯域限度に反した加入者に対応するケーブル インターフェイスの MAC (ハードウェア) アドレス
ccqmEnfRuleViolateRuleName	DisplayString	この加入者に対応付けられた強制ルールの名前。この値を ccqmCmtsEnfRuleName 属性と比較することによって、加入者に割り当てられた強制ルールを調べることができます。
ccqmEnfRuleViolateByteCount	Unsigned32	モニタ期間中に加入者が使用した総キロバイト数(このカウンタは、強制ルールが中止され、再開されるたびにリセットされます)
ccqmEnfRuleViolateLastDetectTime	DateAndTime	加入者が強制ルールに違反していることを Cisco CMTS が判別したときのタイムスタンプ
ccqmEnfRuleViolatePenaltyExpTime	DateAndTime	この加入者のペナルティ期間が満了するときのタイムスタンプ。Cisco CMTS ルータのオペレータがペナルティ期間を手動で削除しないかぎり、加入者はこの日時になるまで、ペナルティ期間が課せられます。この日時になると、元の QoS プロファイルが回復されます。加入者の強制ルールに強制 QoS プロファイルが含まれていなかった場合、この属性は 0 です。

強制ルールに違反した加入者への通知

ユーザが強制ルールに指定された帯域限度に反したときに、通知を送信するように Cisco CMTS を設定するには、`ccqmEnfRuleViolateNotifEnable` オブジェクトを `True` に設定します。デフォルトは `False` (通知を送信しない) です。表 A-14 に、各通知で送信される属性 (`ccqmEnfRuleViolateTable` テーブルで定義) を示します。

表 A-14 `ccqmEnfRuleViolateNotification` の属性

属性	タイプ	説明
<code>ccqmEnfRuleViolateMacAddr</code>	MacAddress	QoS の帯域限度に反した加入者に対応するケーブル インターフェイスの MAC (ハードウェア) アドレス
<code>ccqmEnfRuleViolateRuleName</code>	DisplayString	この加入者に対応付けられた強制ルールの名前。この値を <code>ccqmCmtsEnfRuleName</code> 属性と比較することによって、加入者に割り当てられた強制ルールを調べることができます。
<code>ccqmEnfRuleViolatePenaltyExpTime</code>	DateAndTime	この加入者のペナルティ期間が満了するときのタイムスタンプ。Cisco CMTS ルータのオペレータがペナルティ期間を手動で削除しないかぎり、加入者はこの日時になるまで、ペナルティ期間が課せられます。この日時になると、元の QoS プロファイルが回復されます。加入者の強制ルールに強制 QoS プロファイルが含まれていなかった場合、この属性は 0 です。
<code>ccqmEnfRuleViolateByteCount</code>	Unsigned32	モニタ期間中に加入者が使用した総キロバイト数(このカウンタは、強制ルールが中止され、再開されるたびにリセットされます)

使用状況ベースの課金

CISCO-CABLE-METERING-MIB により、間隔、必須メータリング情報の量、収集サーバの場所または IP アドレス、およびメータリング ファイル名などのメータリング レコード収集を制御するパラメータを設定できます。さらに、メータリング収集の成功または失敗を示す一部の重要な通知を NMS に提供します。

Cisco Cable Modem Termination System (CMTS) の使用状況ベースの課金機能は、加入者のアカウントおよび課金情報を Subscriber Account Management Interface Specification (SAMIS) 形式で提供します。SAMIS 形式は、Data-over-Cable Service Interface Specifications (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) Operations Support System Interface (OSSI) 仕様で指定されます。

使用状況ベースの課金機能には、標準ベースでオープンな、DOCSIS ネットワークのトラフィック課金情報の記録および検索へのアプリケーションによるアプローチが用意されています。イーネブルになっている場合、この機能は、ケーブル ネットワークを使用しているケーブル モデムおよび宅内装置 (CPE) に関する次の課金情報を提供します。

- ケーブル モデムの IP および MAC アドレス
- 使用されているサービス フロー(アップストリームとダウンストリーム両方のサービス フローがトラッキングされます)
- ケーブル モデムを使用している CPE デバイスの IP アドレス
- 収集期間にケーブル モデムが受信した (ダウンストリーム) またはケーブル モデムから送信された (アップストリーム) オクテット数およびパケット数の合計
- 加入者の Service Level Agreement (SLA; サービス レベル アグリーメント) で許可される帯域幅レベルを超えたため CMT が廃棄または遅延した、ケーブル モデムのダウンストリーム パケット数の合計

課金記録は、サービス プロバイダが簡単に既存の課金アプリケーションに統合できる、標準テキスト形式で保持されます。サービス プロバイダは、この情報を使用して、通常ベースで SLA 制限を超過しようとするカスタマーだけでなく、サービスをアップグレードする可能性のあるユーザも判別できます。

動作モード

使用状況ベースの課金機能は、次の 2 つのモードで動作します。

- ファイル モード
- ストリーミング モード

ファイル モード

ファイル モードでは、CMTS が課金記録情報を収集し、その課金記録をローカル ファイル システムのファイルに書き込みます。その際、ルータのホスト名と、ファイルが書き込まれたときのタイムスタンプを使用します。次に、リモート アプリケーションが CMTS にログインし、課金アプリケーションがアクセスできる外部サーバに課金記録ファイルを転送します。

リモート アプリケーションは、Secure Copy Protocol (SCP) または Trivial File Transfer Protocol (TFTP) を使用して、ファイルを転送できます。転送が成功すると、リモート アプリケーションは課金記録ファイルを削除し、この削除によって新しいファイルが作成可能であることを示す信号が CMTS に送られます。リモート アプリケーションは、定期的に CMTS にログインして課金記録ファイルを転送することも、課金記録ファイルが使用可能であることをアプリケーションに通知するために CMTS が SNMPv3 トラップを送信するまで待機することもできます。

使用状況ベースの課金機能（ファイルモード）

使用状況ベースの課金機能に対して Cisco CMTS をファイルモードで設定するには、CISCO-CABLE-METERING-MIB のオブジェクト数を設定する必要があります。表 A-15 に、これらの個々のオブジェクト、およびこれらのオブジェクトが必須か、または任意であるかを示します。

表 A-15 ファイルモードに設定される CISCO-CABLE-METERING-MIB オブジェクト

MIB オブジェクト	タイプ	説明
ccmtrCollectionTable ccmtrCollectionType	整数	<p>使用状況ベースの課金機能をイネーブルまたはディセーブルにします。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> 1 = なし。使用状況ベースの課金機能はディセーブルになっています（デフォルト）。 2 = ローカル。使用状況ベースの課金機能はイネーブルで、ファイルモード用に設定されています。 3 = ストリーム。使用状況ベースの課金機能はイネーブルで、ストリーミングモード用に設定されています。 <p>ccmCollectionType を 2（ローカル）に設定し、ファイルモードに対して機能をイネーブルにします。</p>
ccmtrCollectionFilesystem	DisplayString	<p>課金記録ファイルを書き込むファイルシステムを指定します。このオブジェクトの最大長は 25 文字で、ルータ上（slot0、disk1、または flash）の有効なファイルシステムを指定する必要があります。</p> <p> (注) Cisco CMTS は、ルータのホスト名の後にファイルが書き込まれたときのタイムスタンプが続くファイル名を使用して、このファイルシステムに課金記録を書き込みます。</p>
ccmtrCollectionCpeList	TruthValue	<p>(任意) 課金記録のサイズを減らし、パフォーマンスを向上させるために CPE デバイスの IP アドレスが省略されているかどうかを示します。有効な値は、次のとおりです。</p> <ul style="list-style-type: none"> true = CPE 情報が表示されています（デフォルト）。 false = CPE 情報は省略されています。 <p> (注) true に設定されているとき、ケーブルモデムごとに最大 5 つの CPE の IP アドレスが表示されます。</p>
ccmtrCollectionAggregate	TruthValue	<p>(任意) 個々のケーブルモデムのすべての情報が結合されて、1 つの記録になっているかどうかを示します。アップストリームトラフィックおよびダウンストリームトラフィックに対して個別のカウンタが維持されますが、これらのカウンタには、その方向のサービスフローが含まれています。有効な値は、次のとおりです。</p> <ul style="list-style-type: none"> true = 各ケーブルモデムのすべてのサービスフローが、単一の課金記録に集約されます。この設定では、課金記録のサービスフロー ID (SFID) が 0 に設定され、サービスクラス名 (SCN) は空白になります。 false = 各ケーブルモデムの情報が 1 つの課金記録に集約されることはありませんが、代わりに各サービスフローが自身のレコードに記録されます（デフォルト）。

表 A-15 ファイル モードに設定される CISCO-CABLE-METERING-MIB オブジェクト (続き)

MIB オブジェクト	タイプ	説明
ccmtrCollectionSrcIfIndex		課金パケットの送信元インターフェイスを示します。 ccmtrCollectionType がローカル local(2) の場合、このインターフェイスの IP アドレスが、課金パケットの CMTS IP アドレスとして使用されます。ccmtrCollectionType が remote(3) の場合、課金パケットの CMTS IP アドレスと同様に、課金パケットの送信元 IP アドレスも、このインターフェイスの IP アドレスに変更されます。 メータリングの送信元インターフェイスを指定するように、メータリング モードを設定する必要があります。

ストリーミング モード

ストリーミング モードでは、CMTS が課金記録情報を収集し、その後定期的に課金記録ファイルを外部サーバのアプリケーションに転送します。このとき、非セキュア TCP 接続または SSL 接続のいずれかが使用されます。外部サーバが成功した転送の確認応答を行うと、CMTS は課金記録ファイルを削除し、新しいファイルの作成を開始します。

CMTS が外部サーバとの接続の確立に失敗した場合、設定に従って 1～3 回の間で接続を再試行します。CMTS が引き続き外部サーバと接続できない場合、CMTS は SNMPv3 トラップを送信し、障害が発生したことを SNMP マネージャに知らせることができます。

ストリーミング モードでは、CMTS が定期的な間隔で課金記録ファイルを転送するように設定します。通常、ケーブル モデム数と CMTS が作成する課金記録ファイルのサイズに従って間隔を選択します。

使用状況ベースの課金機能に対して Cisco CMTS をストリーミング モードで設定するには、CISCO-CABLE-METERING-MIB のオブジェクト数を設定する必要があります。表 A-16 に、これらの個々のオブジェクトと、必須であるか任意であるかを示します。

表 A-16 ストリーミング モードに設定される CISCO-CABLE-METERING-MIB オブジェクト

オブジェクト	タイプ	説明
ccmtrCollectionTable ccmtrCollectionType	整数	使用状況ベースの課金機能をイネーブルまたはディセーブルにします。有効な値は次のとおりです。 <ul style="list-style-type: none"> 1 = なし。使用状況ベースの課金機能はディセーブルになっています (デフォルト)。 2 = ローカル。使用状況ベースの課金機能はイネーブルで、ファイル モード用に設定されています。 3 = ストリーム。使用状況ベースの課金機能はイネーブルで、ストリーミング モード用に設定されています。 ccmCollectionType を 3 (ストリーム) に設定し、ストリーミング モードに対して機能をイネーブルにします。
ccmtrCollectionIpAddress	InetAddress	外部収集サーバの IP アドレス。この値は、設定の必要があります。
ccmtrCollectionPort	Unsigned32	課金記録の送信先である、外部収集サーバの TCP ポート番号。有効な値の範囲は 0～65535 ですが、ポートを周知の範囲 0～1024 に指定しないでください。この値は、設定の必要があります。

表 A-16 ストリーミング モードに設定される CISCO-CABLE-METERING-MIB オブジェクト (続き)

オブジェクト	タイプ	説明
 <p>(注) ccmCollectionIpAddress と ccmCollectionPort を 2 回設定して、プライマリ収集サーバとセカンダリ収集サーバを指定できます。</p>		
ccmtrCollectionIpAddrType	InetAddressType	(任意) 収集サーバに使用されている IP アドレスのタイプ。有効な値は ipv4 だけで、これがデフォルト値です。
ccmCollectionInterval	Unsigned32	(任意) 課金記録が外部サーバにストリーミングされる頻度を分単位で指定します。有効な範囲は 15 ~ 1440 分 (24 時間) です。デフォルトは 30 分です (最低の間隔は 30 分にするをお勧めします)。
ccmtrCollectionRetries	Unsigned32	(任意) セカンダリ サーバ (設定されている場合) を使用して障害に関する SNMP トラップを送信する前に、CMTS が外部サーバとのセキュアな接続を確立するための再試行回数を指定します。 <i>n</i> の有効な範囲は 0 ~ 5、デフォルトは 0 です。
 <p>(注) SNMP コマンドでストリーミング モードに使用状況ベースの課金を設定するとき、ccmCollectionInterval パラメータと ccmCollectionRetries パラメータは任意ですが、これらのパラメータは、CLI コマンドで機能を設定するときには必須です。</p>		
ccmtrCollectionSecure	TruthValue	(任意) Cisco CMTS が、外部サーバの課金アプリケーションと接続するとき、SSL 接続を使用するかどうかを指定します。有効な値は次のとおりです。 <ul style="list-style-type: none"> • true(1) = Cisco CMTS は SSL 接続を使用します。このオプションは、Baseline Privacy Interface (BPI) 暗号化をサポートする CMTS ソフトウェア イメージでのみ使用できます。 • false(2) = Cisco CMTS は非暗号化 TCP 接続を使用します。これがデフォルト値です。
ccmtrCollectionCpeList	TruthValue	(任意) 課金記録のサイズを減らし、パフォーマンスを向上させるために CPE デバイスの IP アドレスが省略されているかどうかを示します。有効な値は、次のとおりです。 <ul style="list-style-type: none"> • true = CPE 情報が表示されています (デフォルト)。 • false = CPE 情報は省略されています。  <p>(注) true に設定されているとき、ケーブル モデムごとに最大 5 つの CPE の IP アドレスが表示されます。</p>
ccmtrCollectionAggregate	TruthValue	(任意) 個々のケーブル モデムのすべての情報が結合されて、1 つの記録になっているかどうかを示します。アップストリーム トラフィックおよびダウンストリーム トラフィックに対して個別のカウンタが維持されますが、これらのカウンタには、その方向のサービス フローが含まれています。有効な値は、次のとおりです。 <ul style="list-style-type: none"> • true = 各ケーブル モデムのすべてのサービス フローが、単一の課金記録に集約されます。この設定では、課金記録のサービス フロー ID (SFID) が 0 に設定され、サービス クラス名 (SCN) は空白になります。 • false = 各ケーブル モデムの情報が 1 つの課金記録に集約されることはありませんが、代わりに各サービス フローが自身のレコードに記録されます (デフォルト)。

Cisco CMTS の従量制課金機能の使用方法の詳細については、次の URL を参照してください。

http://www.cisco.com/en/US/products/hw/cable/ps2217/products_feature_guide09186a00801ef1d7.html

Cisco Unique Device Identifier の識別

UDI 検索を使用するには、使用しているシスコ製品が UDI 対応になっている必要があります。UDI 対応のシスコ製品は、5 つの必須 Entity MIB オブジェクトをサポートします。5 つの Entity MIB v2 (RFC-2737) オブジェクトは、次のとおりです。

- entPhysicalName
- entPhysicalDescr
- entPhysicalModelName
- entPhysicalHardwareRev
- entPhysicalSerialNum

show inventory コマンドが使用可能になっていることもありますが、このコマンドを UDI 対応でないデバイスで使用しても、出力は作成されません。

UDI 検索機能を使用する前に、次の概念を理解しておく必要があります。

UDI の概要 — 各識別可能製品は、Entity MIB (RFC-2737) およびサポート マニュアルで定義されているとおり、エンティティです。シャーシなど一部のエンティティには、スロットなどのサブエンティティがあります。イーサネットスイッチが、スタックのようなスーパーエンティティのメンバーになっていることがあります。注文可能なシスコのエンティティの多くは、UDI が割り当てられた状態で出荷されます。

UDI 情報は、物理ハードウェア デバイスに貼られたラベルに印刷されていると同時に、リモート検索を簡単にするために、デバイス上に電子的に保存されています。

UDI は次の要素で構成されます。

- 製品 ID (PID) — PID は、製品を注文するときの名前です。従来より製品名または部品番号と呼ばれています。これは、正確な交換部品を注文するために使用する ID です。
- バージョン ID (VID) — VID は製品のバージョンです。製品が改定されるたびに、VID は 1 つずつ増えていきます。VID は、製品の変更通知を管理する業界ガイドラインである Telcordia GR-209-CORE から派生した、厳密なプロセスに従って増えていきます。
- シリアル番号 (SN) — SN は、製品のベンダー固有通し番号です。製造された各製品には、工場ですべて割り当てられた一意のシリアル番号が付けられていて、この番号を使用側が変更することはできません。この番号により、製品の個別で特定のインスタンスを識別できます。

DOCS-DSG-IF-MIB 検証の要件

検証では、設定が必要な必須パラメータと一貫性ルールを説明し、データが設定された場合、データもこれらのルールに従う必要があります。

ここでは、DOCS-DSG-IF-MIB に関連するエージェントの実装を説明します。従うべきルールは次のとおりです。

- DOCS-DSG-IF-MIB にある MIB テーブルのすべてのインデックスは、相互運用性を持つように、インデックス範囲が SNMP および CLI の間で一貫させておくため、1 ～ 65535 の範囲内になっている必要があります。

RowStatus 列の有効な値は次のとおりです。

- active — 行は完全に使用可能です。
- notInService — 行は完全ですが、アクティブではありません。
- notReady — 行は失われていて、アクティブに設定するときに列が必須です。
- createAndGo — 行を作成し、単一 Set 要求でアクティブになっています。
- createAndWait — 行を作成しますが、アクティブかされるまで notInService または notReady ステータスに保持されます。
- destroy — 行を削除します。

表 A-17 に、DOCS-DSG-IF-MIB にある各 MIB テーブルの機能を示します。

表 A-17 DOCS-DSG-IF-MIB テーブルの機能

MIB テーブル/オブジェクト	最大アクセス	機能
dsgIfClassifierTable	読み取り — 作成	値 : <ul style="list-style-type: none"> • CreateAndGo • CreateAndWait • 変更可能 • 破棄
dsgIfTunnelTable	読み取り — 作成	値 : <ul style="list-style-type: none"> • CreateAndGo • CreateAndWait • 変更可能 • 破棄
dsgIfTunnelGrpToChannelTable	読み取り — 作成	値 : <ul style="list-style-type: none"> • CreateAndGo • CreateAndWait • 変更可能 • 破棄
dsgIfDownstreamTable	読み取り — 書き込み	変更可能
dsgIfClientIdTable	読み取り — 作成	値 : <ul style="list-style-type: none"> • CreateAndGo • 破棄

表 A-17 DOCS-DSG-IF-MIB テーブルの機能 (続き)

MIB テーブル/オブジェクト	最大アクセス	機能
dsgIfVendorParamTable	読み取り — 作成	値： <ul style="list-style-type: none">• CreateAndGo• 破棄
dsgIfChannelListTable	読み取り — 作成	値： <ul style="list-style-type: none">• CreateAndGo• 破棄
dsgIfTimerTable	読み取り — 作成	値： <ul style="list-style-type: none">• CreateAndGo• 破棄

