Cisco NCS 560 シリーズ ルータ（IOS XR リリース 6.6.x）システム管理コンフィギュレーション ガイド

初版：2019 年 5 月 30 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【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、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、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。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.

目 次

第 1 章 フレキシブル コマンドライン インターフェイスの設定 1

フレキシブル CLI 設定グループ 1

フレキシブル設定の制約事項 1

設定グループの構成 3

シンプルな設定グループ：例 4

異なる場所に適用された設定グループ：例 5

設定グループの構成の確認 6

設定グループの正規表現 7

正規表現を使用した設定例 14

正規表現を使用した設定グループ：例 14

正規表現を使用した設定グループの継承：例 16

レイヤ 2 トランスポート設定グループ：例 18

設定グループの優先順位：例 18

設定グループへの変更は自動的に継承される：例 19

第 2 章 ネットワーク タイム プロトコルの設定 21

Cisco IOS XR ソフトウェアで NTP を実装するための前提条件 21

NTP の実装について 21

NTP-PTP インターワーキング 23

Poll-Based アソシエーションの設定 24

Broadcast-Based NTP アソシエーション 26

NTP アクセス グループの設定 29

NTP 認証の設定 30

特定のインターフェイス上の NTP サービスのディセーブル化 32

NTP パケットの送信元 IP アドレスの設定 34
正規の NTP サーバとしてのシステムの設定 36
NTP-PTP インターワーキングの設定 37
ハードウェア クロックの更新 39
外部基準クロックのステータスの確認 40
例 41
NTP の実装の設定例 42
VRF インターフェイス内での NTP サーバの設定 45

第 3 章 管理性の設定 49
XML の管理機能について 49
管理機能の設定方法 50
XML エージェントの設定 50
管理機能の設定例 51
XML エージェントでの VRF のイネーブル化：例 51

第 4 章 オブジェクト トラッキングの設定 53
オブジェクト トラッキングの設定 53
オブジェクト トラッキングの実装の前提条件 53
オブジェクト トラッキングについて 54
オブジェクト トラッキングの実装方法 54
インターフェイスのライン プロトコル ステートのトラッキング 54
IP ルートの到達可能性のトラッキング 56
オブジェクト リストに基づくトラッキングの設定 58
オブジェクト リストに基づくトラッキングの設定：しきい値の割合 60
オブジェクト リストに基づくトラッキングの設定：しきい値の重み 62
IPSLA の到達可能性のトラッキング 64
オブジェクト トラッキングの設定例 65

第 5 章 物理端末および仮想端末の設定 69
物理端末と仮想端末を実装するための前提条件 69

物理端末および仮想端末の実装について 69
ライン テンプレート 69
ライン テンプレート コンフィギュレーション モード 70
ライン テンプレート ガイドライン 70
端末の識別 71
VTY プール 71
Cisco IOS XR ソフトウェアでの物理および仮想端末の実装方法 72
テンプレートの変更 72
VTY プールの作成および変更 73
端末および端末セッションのモニタリング 75
物理および仮想端末の実装の設定例 77

第 6 章 簡易ネットワーク管理プロトコルの設定 81
SNMP の実装の前提条件 81
Cisco IOS XR ソフトウェアでの SNMP の使用に関する制約事項 81
SNMP の実装について 82
SNMP 機能の概要 82
SNMP マネージャ 82
SNMP エージェント 82
MIB 82
SNMP バージョン 83
SNMPv1、SNMPv2c、および SNMPv3 の比較 84
SNMPv1、SNMPv2、SNMPv3 のセキュリティ モデルおよびセキュリティ レベル 85
SNMPv3 の利点 86
SNMPv3 のコスト 87
ユーザベースのセキュリティ モデル 87
View-Based Access Control Model 88
SNMP の IP precedence および DSCP サポート 88
サブスクライバ セッションでのセッション MIB のサポート 89
SNMP 通知 89
セッション タイプ 90

Cisco IOS XR ソフトウェアでの SNMP の実装方法 90
SNMPv3 の設定 90
SNMPv3 の設定：例 92
SNMP トラップ通知の設定 97
トラップ通知の設定：例 98
SNMP エージェントの連絡先、場所、およびシリアル番号の設定 99
SNMP エージェント パケットの最大サイズの定義 100
通知操作値の変更 100
IP precedence および DSCP 値の設定 101
SNMP トラフィックの IP precedence 値の設定：例 102
SNMP トラフィックの IP DSCP 値の設定：例 102
SNMP コンテキスト マッピングの表示 102
パケット損失のモニタリング 103
維持する MIB データの設定 104
インターフェイスのサブセットに対する linkUp および linkDown トラップの設定 105

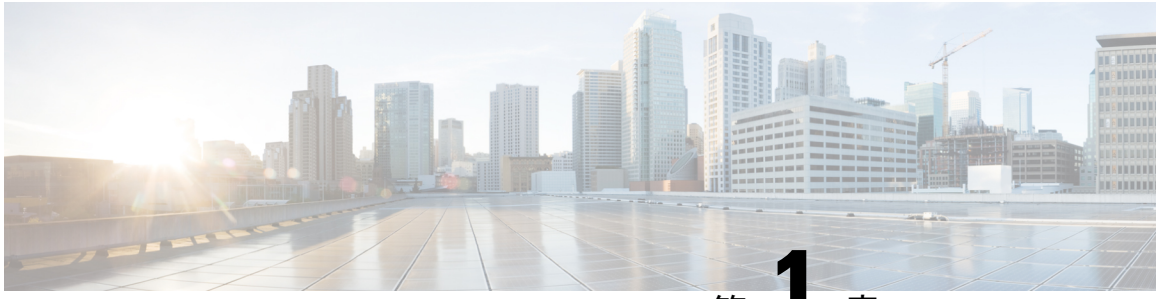
第 7 章 **定期的な MIB データの収集および転送の設定** 109
定期的な MIB データの収集および転送の前提条件 109
定期的な MIB データの収集および転送に関する情報 109
SNMP のオブジェクトとインスタンス 109
バルク統計情報オブジェクト リスト 110
バルク統計情報スキーマ 110
バルク統計情報転送オプション 110
定期的な MIB データの収集および転送の利点 111
定期的な MIB データの収集および転送の設定方法 111
バルク統計情報オブジェクト リストの設定 111
バルク統計情報スキーマの設定 112
バルク統計情報転送オプションの設定 114
定期的な MIB データの収集および転送：例 117

第 8 章 **Cisco Discovery Protocol の設定** 119

CDP の実装の前提条件 119
CDP の実装について 119
CDP の実装方法：Cisco IOS XR ソフトウェア 121
CDP のイネーブル化 121
CDP デフォルト設定の変更 122
CDP のモニタリング 123
CDP の実装の設定例 124

第 9 章 スマート ライセンスを使用したライセンスの設定 127
スマート ライセンスとは? 127
スマート ライセンスの仕組み 128
スマート ライセンスの導入オプション 129
Call Home について 131
サポート対象の柔軟な消費モデル ライセンス 131
スマート ライセンスを使用したライセンスの設定 132
デバイスの登録とアクティブ化 132
スマート ライセンス設定の確認 133
スマート ライセンス登録の更新 136
スマート ライセンスの登録解除 136
スマート ライセンスのワークフロー 137
ライセンス、製品インスタンス、および登録トークン 137
仮想アカウント 138
コンプライアンス レポート 138

第 10 章 ゼロ タッチ プロビジョニングの設定 141
手動 ZTP 呼び出し 143
データ ポートでの認証 144
ZTP ブートスクリプト 145
ZTP ユーティリティ 146
例 147



第 1 章

フレキシブルコマンドラインインターフェイスの設定

このモジュールでは、フレキシブル コマンド ライン インターフェイス（CLI）の設定グループを構成および使用する方法について説明します。

- フレキシブル CLI 設定グループ （1 ページ）
- フレキシブル設定の制約事項 （1 ページ）
- 設定グループの構成 （3 ページ）
- 設定グループの構成の確認 （6 ページ）
- 設定グループの正規表現 （7 ページ）

フレキシブル CLI 設定グループ

フレキシブル コマンド ライン インターフェイス（CLI）の設定グループは、設定グループ内に一連の構成ステートメントを定義し、このグループをルータ設定ツリーの複数の階層レベルに適用することにより、反復構成を最小限に抑える機能を提供します。

フレキシブルCLI設定グループは、階層内でグループが適用されている場所に基づいて、設定ツリーの複数のサブモードで一致が確認される正規表現を使用します。設定サブモードで一致が見つかった場合、そのグループに定義されている対応する構成が、一致するサブモード内で継承されます。

フレキシブル CLI 設定グループには、自動継承機能もあります。自動継承とは、CLI 設定グループに対して行われた変更が、その階層レベルで適用グループを持つ一致するサブモードの構成に自動的に適用されることを意味します。これにより、フレキシブルCLI設定グループを適用した場所に応じて、構成の変更または追加を一度行い、複数の場所に自動的に適用することができます。

フレキシブル設定の制約事項

フレキシブル設定グループを使用する場合は、次の制約事項に注意してください。

- フレキシブルCLI設定グループは管理構成ではサポートされず、対応する適用グループは管理構成ではサポートされません。
- 設定グループ内の事前設定されたインターフェイスの使用はサポートされていません。
- 設定グループをサポートしているイメージからサポートしていないイメージにダウングレードすることはサポートされていません。
- アクセス リスト、QoS およびルート ポリシー設定は、設定グループの使用をサポートしていません。次のような構成は有効ではありません。

```
group g-not-supported
 ipv4 access-list ...
 !
 ipv6 access-list ...
 !
 ethernet-service access-list ...
 !
 class-map ...
 !
 policy-map ...
 !
 route-policy ...
 !
end-group
```

ただし、次の例に示すような構成を参照できます。

```
group g-reference-ok
 router bgp 6500
  neighbor 7::7
   remote-as 65000
   bfd fast-detect
   update-source Loopback300
   graceful-restart disable
   address-family ipv6 unicast
       route-policy test1 in
    route-policy test2 out
    soft-reconfiguration inbound always
   !
  !
 !
 interface Bundle-Ether1005
   bandwidth 10000000
   mtu 9188
      service-policy output input_1
   load-interval 30
 !
end-group
```

- 一部の正規表現はグループ内ではサポートされていません。たとえば、‘?’、‘|’ および ‘$’ はグループ内ではサポートされていません。また、/d や /w などの文字はサポートされていません。
  - 正規表現内で複数の一致式を表す選択演算子「|」はサポートされていません。たとえば、次の式はサポートされていません。

`Gig.*|Gig.*\..*`：ギガビット イーサネット インターフェイスまたはギガビット イーサネット サブインターフェイスでの照合時。

`Gig.*0/0/0/[1-5]|Gig.*0/0/0/[10-20]`：Gig.*0/0/0/[1-5] または Gig.*0/0/0/[10-20] での照合時。

`'TenGigE.*|HundredGigE.*`：TenGigE.* または HundredGigE.* での照合時。

- **location** キーワードのノード識別子を必要とするコマンドはサポートされていません。たとえば、次の構成はサポートされません。

```
lpts pifib hardware police location 0/RP0/CPU0
```

- 同じ構成に対する設定グループ内の重複する正規表現はサポートされていません。次に例を示します。

```
group G-INTERFACE
interface 'gig.*a.*'
  mtu 1500
!
interface 'gig.*e.* '
  mtu 2000
!
end-group

interface gigabitethernet0/0/0/* ---- where * is 0 to 31
  apply-group G-INTERFACE
```

`interface GigabitEthernet0/0/0/*` 設定が `mtu 1500` または `mtu 2000` を継承するかどうかを判別できないため、この設定は許可されません。設定グループの両方の式は、GigabitEthernet0/0/0/* と一致します。

- 1 つの apply-group コマンドでは、最大 8 つの設定グループが許可されます。

設定グループの構成

設定グループには、ルータ設定ツリーの複数の階層レベルで使用できる一連の構成ステートメントが含まれています。設定グループで正規表現を使用すると、複数のインスタンスに適用できる汎用コマンドを作成できます。

設定グループを作成して使用するには、このタスクを使用します。

（注） フレキシブル CLI 設定は、XML インターフェイスでは利用できません。

手順

ステップ 1 **configure**

ステップ 2 **group** *group-name*

例：

```
RP/0/RP0/CPU0:router(config)# group g-interf
```

設定グループの名前を指定し、グループを定義するためのグループ コンフィギュレーション モードを開始します。*group-name* 引数は最大 32 文字で、特殊文字は使用できません。

ステップ 3 グローバル コンフィギュレーション モードから開始して、設定コマンドを入力します。インターフェイス名やその他の変数インスタンスには正規表現を使用します。

例：

```
RP/0/RP0/CPU0:router(config)# group g-interf
RP/0/RP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if)# mtu 1500
```

この設定グループに含める構成ステートメントを指定します。

正規表現の使用に関する詳細については、正規表現を使用した設定グループの継承：例 （16 ページ） を参照してください。この例は、すべてのギガビット イーサネット インターフェイスに適用されます。

ステップ 4 **end-group**

例：

```
RP/0/RP0/CPU0:router(config-GRP-if)# end-group
```

設定グループの設定を完了し、グローバル コンフィギュレーション モードを終了します。

ステップ 5 **apply-group**

例：

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet0/2/0/0
RP/0/RP0/CPU0:router(config-if)# apply-group g-interf
```

グループが適用されている場所に適用可能なルータ構成に設定グループの構成を追加します。グループは複数の場所に適用することができ、その影響は場所とコンテキストによって異なります。

グループ g-interf からの MTU 値は、インターフェイス TenGigE0/11/0/0 に適用されます。このグループがグローバル コンフィギュレーション モードで適用される場合、この MTU 値は、MTU 値が設定されていないすべてのギガビット イーサネット インターフェイスによって継承されます。

シンプルな設定グループ：例

次に、設定グループを使用してグローバル設定をシステムに追加する例を示します。

```
RP/0/RP0/CPU0:router(config)# group g-logging
RP/0/RP0/CPU0:router(config-GRP)# logging trap notifications
RP/0/RP0/CPU0:router(config-GRP)# logging console debugging
RP/0/RP0/CPU0:router(config-GRP)# logging monitor debugging
RP/0/RP0/CPU0:router(config-GRP)# logging buffered 10000000
RP/0/RP0/CPU0:router(config-GRP)# end-group

RP/0/RP0/CPU0:router(config)# apply-group g-logging
```

この設定がコミットされると、g-logging 設定グループに含まれるすべてのコマンドがコミットされます。

異なる場所に適用された設定グループ：例

設定グループは異なる場所に適用でき、その影響は適用されるコンテキストによって異なります。次の設定グループを考えてみましょう。

```
RP/0/RP0/CPU0:router(config)# group g-interfaces
RP/0/RP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if)# mtu 1500
RP/0/RP0/CPU0:router(config-GRP-if)# exit
RP/0/RP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if)# mtu 1000
RP/0/RP0/CPU0:router(config-GRP-if)# exit
RP/0/RP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if)# mtu 2000
RP/0/RP0/CPU0:router(config-GRP-if)# end-group
```

このグループはギガビットイーサネットインターフェイスに適用でき、各インスタンスでは、適用可能な MTU が適用されます。たとえば、次の例では、ギガビット イーサネット インターフェイスの MTU が 1000 に設定されています。

```
RP/0/RP0/CPU0:router(config)# interface TenGigE0/11/0/0
RP/0/RP0/CPU0:router(config-if)# apply-group g-interfaces
RP/0/RP0/CPU0:router(config-if)# ipv4 address 2.2.2.2 255.255.255.0
```

次の例では、ギガビット イーサネット インターフェイスの MTU が 1500 に設定されています。

```
RP/0/RP0/CPU0:router(config)# interface TenGigE0/11/0/0
RP/0/RP0/CPU0:router(config-if)# apply-group g-interfaces
RP/0/RP0/CPU0:router(config-if)# ipv4 address 3.3.3.3 255.255.255.0
```

どちらの場合も同じ設定グループが使用されていますが、適用可能な構成ステートメントのみが使用されています。

設定グループの構成の確認

このタスクを使用して、設定グループを使用したルータの設定を確認します。

手順

	コマンドまたはアクション	目的

ステップ 1	**show running-config group** [*group-name*]
	例：
	`RP/0/RP0/CPU0:router# show running-config group`
	`group g-int-ge`
	` interface 'GigabitEthernet.*'`
	`  mtu 1000`
	`  negotiation auto`
	` !`
	`end-group`
		特定のまたはすべての構成済み設定グループの内容を表示します。

ステップ 2	**show running-config**
	例：
	`RP/0/RP0/CPU0:router# show running-config`
	`group G-INTERFACE-MTU`
	` interface 'GigabitEthernet.*'`
	`  mtu 1500`
	` !`
	`end-group`
	`interface interface TenGigE0/11/0/0`
	` apply-group G-INTERFACE-MTU`
	`!`
	`interface interface TenGigE0/11/0/1`
	` apply-group G-INTERFACE-MTU`
	` mtu 2000`
	`!`
		実行コンフィギュレーションを表示します。適用されているグループが表示されます。これらの設定グループが実際の構成に影響を与えるかどうかに関しては示されません。この例では、グループ G-INTERFACE-MTU は interface TenGigE0/11/0/1 に適用されますが、設定された MTU 値は 1500 ではなく 2000 です。これは、コマンド **mtu 2000** がインターフェイス上で直接設定されている場合に発生します。実際の構成と設定グループの構成が同じであれば、実際の構成が優先されます。

ステップ 3	**show running-config inheritance**
	例：
	`RP/0/RP0/CPU0:router# show running-config inheritance`
	`.`
	`.`
	`group G-INTERFACE-MTU`
	` interface 'GigabitEthernet.*'`
	`  mtu 1500`
	` !`
	`end-group`
		設定グループが適用されている継承された構成を表示します。

	コマンドまたはアクション	目的
	. . interface interface TenGigE0/11/0/0 ## Inherited from group G-INTERFACE-MTU mtu 1500 ! interface interface TenGigE0/11/0/1 mtu 2000 ! . .	
ステップ 4	**show running-config interface x/y/z inheritance** *detail* 例： RP/0/RP0/CPU0:router# **show running-config interface** interface TenGigE0/11/0/0 **inheritance detail** interface interface TenGigE0/11/0/0 ## Inherited from group G-INTERFACE-MTU mtu 1500	特定のコンフィギュレーション コマンドに対する継承された構成を表示します。

設定グループの正規表現

正規表現は、設定グループを幅広く適用可能にするために設定グループで使用されます。Portable Operating System Interface for UNIX（POSIX）1003.2 正規表現は、構成ステートメントの名前でサポートされています。正規表現を区切るには、単一引用符を使用する必要があります。

（注） すべての POSIX 正規表現がサポートされているわけではありません。

インターフェイス識別子の正規表現

設定グループは、正確なインターフェイス識別子を受け入れません。設定グループに適用可能なインターフェイスのグループを識別するには、正規表現を使用する必要があります。正規表現 '.*' は使用できません。インターフェイス識別子の正規表現は、一義的な単語で始まり、その後に正規表現を続けなければなりません。たとえば、ギガビットイーサネットインターフェイスを設定するには、正規表現 'GigabitEthernet.*' を使用します。

ルータ設定で使用可能なインターフェイス タイプのリストを表示するには、設定グループ プロンプトで **interface ?** と入力します。

```
RP/0/RP0/CPU0:router(config-GRP)# interface ?
```

```
  ATM              'RegExp': ATM Network Interface(s)
  BVI              'RegExp': Bridge-Group Virtual Interface
  Bundle-Ether     'RegExp': Aggregated Ethernet interface(s)
  GigabitEthernet  'RegExp': GigabitEthernet/IEEE 802.3 interface(s)
  IMA              'RegExp': ATM Network Interface(s)
  Loopback         'RegExp': Loopback interface(s)
  MgmtEth          'RegExp': Ethernet/IEEE 802.3 interface(s)
  Multilink        'RegExp': Multilink network interface(s)
  Null             'RegExp': Null interface
  PW-Ether         'RegExp': PWHE Ethernet Interface
  PW-IW            'RegExp': PWHE VC11 IP Interworking Interface
  Serial           'RegExp': Serial network interface(s)
  tunnel-ip        'RegExp': GRE/IPinIP Tunnel Interface(s)
  tunnel-mte       'RegExp': MPLS Traffic Engineering P2MP Tunnel interface(s)
  tunnel-te        'RegExp': MPLS Traffic Engineering Tunnel interface(s)
  tunnel-tp        'RegExp': MPLS Transport Protocol Tunnel interface
```

（注） インターフェイスのタイプを一意なものにするのに十分な文字数のみを入力するよう要求されますが、フレーズ全体を入力することをお勧めします。正規表現で使用されるすべてのインターフェイス タイプでは、大文字と小文字が区別されます。

サブインターフェイスを指定するには、式の前に\.という文字を付けます（バックスラッシュピリオド）。たとえば、すべてのギガビット イーサネット サブインターフェイスを設定するには、`interface 'GigabitEthernet.*\..*'` を使用します。

次の例に示すように、レイヤ2トランスポートインターフェイスまたはポイントツーポイントインターフェイスを指定できます。

```
group g-l2t
  interface 'Gi.*\..*' l2transport
.
.
end-group
group g-ptp
  interface 'Gi.*\..*' point-to-point
.
.
end-group
```

OSPF 設定の正規表現

正確なルータ プロセス名と OSPF 領域は使用できません。プロセス名または OSPF 領域のグループを指定するには、正規表現を使用する必要があります。OSFP 領域がスカラー値またはIP アドレスのいずれかになるように指定するには、次の例のように正規表現 '.*' を使用します。

```
group g-ospf
router ospf '.*'
area '.*'
mtu-ignore enable
!
!
```

```
end-group
```

OSPF 領域を IP アドレスにする必要があることを指定するには、次の例のように式 '\.' を使用します。

```
group g-ospf-ipaddress
router ospf '.*\..*\..*\..*'
area '.*'
passive enable
!
!
 end-group
```

OSPF 領域をスカラー値にする必要があることを指定するには、次の例のように式 '1.*' を使用します。

```
group g-ospf-match-number
router ospf '.*'
area '1.*'
passive enable
!
!
end-group
```

BGP AS の正規表現

正確な BGP AS 値は、設定グループでは使用できません。形式 X.Y などで AS プレーン形式または AS ドット形式を指定するには正規表現を使用します。AS プレーン形式のインスタンスを照合するには、単純な正規表現を使用します。AS ドット形式のインスタンスを照合するには、次の例に示すように、ドットで区切られた 2 つの正規表現を使用します。

```
group g-bgp
router bgp '*'.'*'
address-family ipv4 unicast
!
!
end-group
```

ANCP の正規表現

正確なアクセス ノード制御プロトコル（ANCP）送信者名識別子は設定グループで使用できません。送信者名の引数は IP アドレスまたは MAC アドレスのどちらにでもできるので、正規表現ではどちらが使用されているかを指定する必要があります。IP アドレスは `'.*\..*\..*\..*'` と指定します。MAC アドレスは `'.*\..*\..*'` と指定します。

ユニフォーム タイプへの解決

正規表現は、ユニフォーム タイプに解決されなければなりません。次は、不正な正規表現の例です。

```
group g-invalid
```

```
 interface '.*'
  bundle port-priority 10
 !
interface '.*Ethernet.*'
  bundle port-priority 10
 !
end-group
```

この例では、**bundle** コマンドはインターフェイス タイプ GigabitEthernet ではサポートされていますが、インターフェイス タイプ 'FastEthernet' ではサポートされていません。正規表現 '.*' および '.*Ethernet.*' は、GigabitEthernet タイプと FastEthernet タイプの両方に一致します。**bundle** コマンドはこれらの両方のインターフェイス タイプには適用できず、ユニフォーム タイプに解決しないので、システムはこの設定を許可しません。

（注） システムが正規表現からどの構成をすべきかを判別できない場合、その式は有効であると見なされません。

（注） インターフェイス識別子の参照時に正規表現 '.*' は使用できません。インターフェイス識別子の正規表現は、一義的な単語で始まり、その後に正規表現を続けなければなりません。詳細については、この項の「インターフェイス識別子の正規表現」を参照してください。

重複する正規表現

正規表現は、設定グループ内の構成ステートメントの名前で使用されます。これにより、一致する名前に適用された場合に設定による継承が可能になります。正規表現を区切るには、単一引用符が使用されます。同じ構成に対し設定グループ内の重複する正規表現は許可されません。

以下に示す例は、複数の設定グループを作成して適用するプロセスを示しています。

```
RP/0/RP0/CPU0:router(config)#group FB_flexi_snmp
RP/0/RP0/CPU0:router(config-GRP)# snmp-server vrf '.*'
RP/0/RP0/CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 traps version 2c group_1
RP/0/RP0/CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 informs version 2c group_1
RP/0/RP0/CPU0:router(config-GRP-snmp-vrf)# context group_1

RP/0/RP0/CPU0:router(config-GRP-snmp-vrf)#
RP/0/RP0/CPU0:router(config-GRP-snmp-vrf)#commit

RP/0/RP0/CPU0:router(config-GRP-snmp-vrf)#root
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config)#snmp-server vrf vrf1
RP/0/RP0/CPU0:router(config-snmp-vrf)#snmp-server vrf vrf10
RP/0/RP0/CPU0:router(config-snmp-vrf)#!
RP/0/RP0/CPU0:router(config-snmp-vrf)#snmp-server vrf vrf100
RP/0/RP0/CPU0:router(config-snmp-vrf)#
RP/0/RP0/CPU0:router(config-snmp-vrf)#commit

RP/0/RP0/CPU0:router(config-snmp-vrf)#root
```

```
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config)#apply-group FB_flexi_snmp
RP/0/RP0/CPU0:router(config)#do sh running-config group
group FB_flexi_snmp
 snmp-server vrf '.*'
  host 1.1.1.1 traps version 2c group_1
  host 1.1.1.1 informs version 2c group_1
  context group_1
 !
end-group
apply-group FB_flexi_snmp
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
RP/0/RP0/CPU0:ios#show running-config inheritance detail


group FB_flexi_snmp
 snmp-server vrf '.*'
  host 1.1.1.1 traps version 2c group_1
  host 1.1.1.1 informs version 2c group_1
  context group_1
 !
end-group
snmp-server vrf vrf1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 traps version 2c group_1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 informs version 2c group_1
 ## Inherited from group FB_flexi_snmp
 context group_1
!
snmp-server vrf vrf10
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 traps version 2c group_1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 informs version 2c group_1
 ## Inherited from group FB_flexi_snmp
 context group_1
!
snmp-server vrf vrf100
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 traps version 2c group_1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 informs version 2c group_1
 ## Inherited from group FB_flexi_snmp
 context group_1
```

次の例は、正規表現を示しています。この例では、`snmp-server vrf '.*'` および `snmp-server vrf '[\w]+` は 2 つの異なる正規表現です。

```
group FB_flexi_snmp
snmp-server vrf '.*'
host 1.1.1.1 traps version 2c group_1
host 1.1.1.1 informs version 2c group_1
context group_1
!
```

```
snmp-server vrf '[\w]+'
host 2.2.2.2 traps version 2c group_2
host 2.2.2.2 informs version 2c group_2
context group_2
!
end-group
```

この個々の正規表現は、次に示すように3つの式`snmp-server vrf vrf1`、`snmp-server vrf vrf10`および`snmp-server vrf vrf100`すべてに結合されます。

```
apply-group FB_flexi_snmp
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
```

設定グループには、正規表現の重複のインスタンスが存在することがあります。このような場合、適用されたときに最も優先順位の高い正規表現がアクティブ化され、継承されます。その正規表現は、最も優先順位の高い辞書式順で最初に来ます。

次の例は、重複する正規表現を使用する方法と、優先順位の高い式がどのように適用されるかを示しています。

```
 group FB_flexi_snmp
 snmp-server vrf '.*'
  host 1.1.1.1 traps version 2c group_1
  host 1.1.1.1 informs version 2c group_1
  context group_1
 !
 snmp-server vrf '[\w]+'
  host 2.2.2.2 traps version 2c group_2
  host 2.2.2.2 informs version 2c group_2
  context group_2
 !
end-group
```

次に示す式は最も優先順位が高いです。

```
  group FB_flexi_snmp
  snmp-server vrf '.*'
  host 1.1.1.1 traps version 2c group_1
  host 1.1.1.1 informs version 2c group_1
```

```
context group_1
```

上記の例は、2 つの異なる正規表現 `snmp-server vrf '.*'` と `snmp-server vrf '[\w]+'` を示しています。

次の式は、これら 2 つの式がどのようにマージされるかを示しています。

```
apply-group FB_flexi_snmp

snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
```

優先順位の低い正規表現の変更は、継承に影響しません。

優先順位が低い（上位ではない）既存の正規表現に加えられた変更は、継承には影響しません。

```
snmp-server vrf '[\w]+'

host 2.2.2.2 traps version 2c group_2
host 2.2.2.2 informs version 2c group_2
context group_2
```

次に示すように、優先順位の高い式が継承されます。

```
group FB_flexi_snmp

snmp-server vrf '.*'

 host 1.1.1.1 traps version 2c group_1

 host 1.1.1.1 informs version 2c group_1

 context group_1
```

グループの優先順位継承の適用

優先順位によって継承が制御されます。

（注） Cisco IOS XR リリース 6.3.1 以降では、コミット全体に必要なすべてのグループ定義があれば、フレキシブル CLI 設定グループ定義、**apply-group** および **exclude-group** コマンドを任意の順序で入力できます。

グループの優先順位の継承を適用すると、フレキシブルな設定グループがグループ間の共通の構成ステートメントを処理できるようになります。複数の設定グループに共通の構成ステートメントがある場合、継承の優先順位は、内部グループに存在する構成ステートメントが外部グループに存在する構成ステートメントよりも優先されます。タイブレーカーの場合、優先順位

は正規表現の辞書式順序に従って割り当てられます。ユーザが定義したコマンドの順序は受け入れられません。

たとえば、設定グループ ONE の構成ステートメントは、別のグループよりも優先されます。設定グループ SEVEN の構成ステートメントは、他のグループに存在しない場合にのみ使用されます。設定グループ内では、継承の優先順位は最長一致です。

```
   apply-group SIX SEVEN
    router ospf 0
   apply-group FOUR FIVE
    area 0
   apply-group THREE
    interface TenGigE0/11/0/0
   apply-group ONE TWO

  !
 !
!
```

上記の例は、2つのシナリオを示しています。最も内側のグループ（**apply-group ONE TWO**）が最も優先順位が高くなります。 ケース 1

最初のシナリオは、どのグループが優先順位を得るかを示しています。この例では、どのグループが異なる設定グループ（共通のものがない異なるグループ）間で適用されるかを示しています。グループ 1（ONE TWO）を適用すると、7 つのグループすべてがインターフェイス `interfaceTenGigE0/11/0/0` に一致し、適用されます。

ケース 2

ここで、すべてが同じ（共通の）構成を持つ場合、グループ1がアクティブになります。つまり、`apply-group ONE TWO` がアクティブになります。グループ 1 が削除されると、グループ 2 がアクティブになります。

正規表現を使用した設定例

正規表現を使用した設定グループ：例

次の例では、正確なインターフェイスに正規表現を使用して、ISIS ルーティング パラメータでギガビットイーサネットインターフェイスを設定するための設定グループの定義を示しています。

```
RP/0/RP0/CPU0:router(config)# group g-isis-gige
RP/0/RP0/CPU0:router(config-GRP)# router isis '.*'
RP/0/RP0/CPU0:router(config-GRP-isis)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-isis-if)# lsp-interval 20
RP/0/RP0/CPU0:router(config-GRP-isis-if)# hello-interval 40
RP/0/RP0/CPU0:router(config-GRP-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-GRP-isis-if-af)# metric 10
RP/0/RP0/CPU0:router(config-GRP-isis-if-af)# end-group
RP/0/RP0/CPU0:router(config)#
```

この設定グループの使用について説明するために、ISIS ルーティング パラメータを使用してこれらのギガビットイーサネットインターフェイスを設定すると仮定します。

```
router isis green
 interface TenGigE0/11/0/0
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
 interface TenGigE0/11/0/1
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
 interface TenGigE0/11/0/2
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
 interface TenGigE0/11/0/3
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
!
```

設定グループを使用してこれらのインターフェイスを設定するには、3 つの方法が考えられます。1 つ目は、次に示すように、インターフェイス設定内でグループを適用することです。

```
router isis green
 interface TenGigE0/11/0/0
     apply-group g-isis-gige
  !
 !
 interface TenGigE0/11/0/1
     apply-group g-isis-gige
  !
 !
 interface TenGigE0/11/0/2
     apply-group g-isis-gige
  !
 !
 interface TenGigE0/11/0/3
     apply-group g-isis-gige
  !
 !
```

この状況では、設定グループを適用するインターフェイスのみが設定を継承します。

設定グループを使用してこれらのインターフェイスを設定する 2 つ目の方法は、次に示すように、**router isis** 設定内に設定グループを適用することです。

```
router isis green
    apply-group g-isis-gige
 interface TenGigE0/11/0/0
 !
 interface TenGigE0/11/0/1
 !
 interface TenGigE0/11/0/2
 !
 interface TenGigE0/11/0/3
 !
!
```

このようにすると、ISIS グリーン設定で設定する他のギガビット イーサネット インターフェイスもこれらの設定を継承します。

設定グループを使用してこれらのインターフェイスを設定する 3 つ目の方法は、次に示すように、グローバル レベルでグループを適用することです。

```
   apply-group g-isis-gige
router isis green
 interface TenGigE0/11/0/0
 !
 interface TenGigE0/11/0/1
 !
 interface TenGigE0/11/0/2
 !
 interface TenGigE0/11/0/3
 !
!
```

この例では、ISIS 用に設定されたすべてのギガビットイーサネットインターフェイスにグループの設定が適用されます。

正規表現を使用した設定グループの継承：例

設定グループより優先されるローカル設定

明示的な設定は、設定グループから適用された設定よりも優先されます。たとえば、次の設定がルータ上で実行されているとします。

```
router ospf 100
 packet-size 1000
!
```

次の設定グループを構成して適用し、設定にコミットします。

```
RP/0/RP0/CPU0:router(config)# group g-ospf
RP/0/RP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf)# nsf cisco
```

```
RP/0/RP0/CPU0:router(config-GRP-ospf)# packet-size 3000
RP/0/RP0/CPU0:router(config-GRP-ospf)# end-group

RP/0/RP0/CPU0:router(config)# apply-group g-ospf
```

結果は事実上次の構成となります。

```
router ospf 100
 packet-size 1000
 nsf cisco
```

明示的なローカル設定が優先されるため、`packet-size 3000` は設定グループから継承されないことに注意してください。

互換性のある構成は継承される

設定グループの構成は、継承されるルータ上の構成と一致する必要があります。構成が一致しない場合は、継承されません。たとえば、次の設定がルータ上で実行されているとします。

```
router ospf 100
 auto-cost disable
!
```

次の設定を構成し、設定にコミットします。

```
RP/0/RP0/CPU0:router(config)# group g-ospf
RP/0/RP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf)# area '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf-ar)# packet-size 2000
RP/0/RP0/CPU0:router(config-GRP-ospf)# end-group

RP/0/RP0/CPU0:router(config)# apply-group g-ospf

RP/0/RP0/CPU0:router(config)# router ospf 200
RP/0/RP0/CPU0:router(config-ospf)# area 1
```

結果は事実上次の構成となります。

```
router ospf 100
  auto-cost disable

router ospf 200
  area 1
  packet-size 2000
```

パケット サイズは、ospf 200 構成によって継承されますが、ospf 100 構成では領域が設定されていないため継承されません。

レイヤ 2 トランスポート設定グループ：例

次に、レイヤ 2 トランスポート サブインターフェイスを使用して設定グループを構成および適用する例を示します。

```
RP/0/RP0/CPU0:router(config)# group g-l2trans-if
RP/0/RP0/CPU0:router(config-GRP)# interface 'TenGigE.*\..*' l2transport
RP/0/RP0/CPU0:router(config-GRP)# mtu 1514
RP/0/RP0/CPU0:router(config-GRP)# end-group

RP/0/RP0/CPU0:router(config)# interface TenGigE0/0/0/0.1 l2transport
RP/0/RP0/CPU0:router(config-if)# apply-group g-l2trans-if
```

この設定がコミットされると、10 ギガビット イーサネット インターフェイス 0/11/0/0.1 は 1514 MTU 値を継承します。これは、10 ギガビット イーサネット インターフェイスの **show running-config inheritence** コマンドで表示される出力です。

```
interface TenGigE0/11/0/0.1 l2transport
 ## Inherited from group g-l2trans-if
 mtu 1514
!
```

設定グループの優先順位：例

同様の構成ステートメントが複数の設定グループに含まれている場合、内部の設定モードで適用されるグループが外部モードで適用されるグループよりも優先されます。次の例では、OSPF にさまざまなコスト値を設定している 2 つの設定グループを示しています。

```
RP/0/RP0/CPU0:router(config)# group g-ospf2
RP/0/RP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf)# area '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf-ar)# cost 2
RP/0/RP0/CPU0:router(config-GRP-ospf-ar)# end-group

RP/0/RP0/CPU0:router(config)# group g-ospf100
RP/0/RP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf)# area '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf-ar)# cost 100
RP/0/RP0/CPU0:router(config-GRP-ospf-ar)# end-group
```

これらの設定グループを次のように適用する場合、g-ospf2 で指定されたコスト 2 は、グループがより内部の設定モードで適用されるため、OSPF エリア 0 によって継承されます。この場合、グループ g-ospf100 の設定は無視されます。

```
RP/0/RP0/CPU0:router(config)# router ospf 0
RP/0/RP0/CPU0:router(config-ospf)# apply-group g-ospf100
RP/0/RP0/CPU0:router(config-ospf)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# apply-group g-ospf2
```

設定グループへの変更は自動的に継承される：例

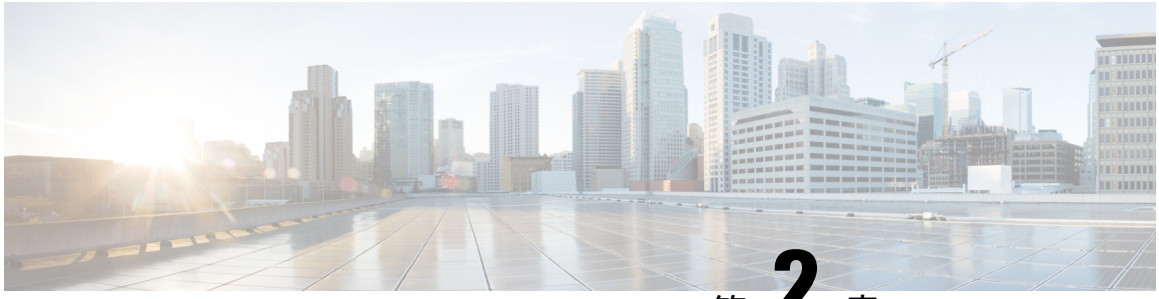
ルータ構成にコミットされて適用されている設定グループに変更を加えると、変更はルータ構成によって自動的に継承されます。たとえば、次の構成がコミットされているとします。

```
group g-interface-mtu
 interface 'GigabitEthernet.*'
  mtu 1500
 !
end-group

interface POS0/0/0/0
 apply-group g-interface-mtu
!
```

ここで、次の例のように設定グループを変更します。

```
RP/0/RP0/CPU0:router(config)# group g-interface-mtu
RP/0/RP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if)# mtu 2000
RP/0/RP0/CPU0:router(config-GRP-if)# end-group
```

この設定グループがコミットされると、インターフェイス GigabitEthernet0/0/0/0 の MTU 設定が自動的に 2000 に更新されます。



第 2 章

ネットワーク タイム プロトコルの設定

ネットワーク タイム プロトコル（NTP）は、ネットワーク内でデバイスの時刻同期を行うように設計されたプロトコルです。Cisco IOS XR ソフトウェア は NTPv4 を実装しています。NTPv4 は以前の NTP バージョンである NTPv3、NTPv2 との後方互換性はありますが、セキュリティ脆弱性のため中止となった NTPv1 との互換性はありません。

- Cisco IOS XR ソフトウェアで NTP を実装するための前提条件 （21 ページ）
- NTP の実装について （21 ページ）
- NTP の実装の設定例 （42 ページ）
- VRF インターフェイス内での NTP サーバの設定 （45 ページ）

Cisco IOS XR ソフトウェアで NTP を実装するための前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

NTP の実装について

NTP を使用すると、分散されたタイム サーバとクライアントの間で時刻が同期されます。同期化により、システム ログ作成時または時間に関するイベントの発生時に、各イベントを関連付けることができます。

NTP ではトランスポート プロトコルとして、ユーザ データグラム プロトコル（UDP）を使用します。NTP の通信はすべて協定世界時（UTC）を使用します。NTP のネットワークでは通常、タイム サーバに接続された電波時計や原子時計など正規の時刻源から時刻を取得します。NTP はこの時刻をネットワーク全体に配信します。NTP はきわめて効率的で、毎分 1 パケット以下で 2 台のマシンを相互に 1 ミリ秒以内に同期します。

NTP では、各マシンが信頼できる時刻源から何 NTP ホップ隔たっているかを表すために「ストラタム」という概念を使用します。「ストラタム 1」タイム サーバには通常、正規の時刻源（電波時計、原子時計、GPS 時刻源など）が直接接続されています。「ストラタム 2」タイム サーバは、「ストラタム 1」タイム サーバから NTP を介して時刻を受信し、それ以降のサーバも続きます。

NTP では、2 つの方法で時刻が間違っている可能性のあるマシンとの同期を回避します。まず、NTP はそれ自身で同期を行わないマシンとの同期を回避します。次に、複数のマシンから報告された時間と大幅に時間が異なっているマシンがある場合、ストラタムの番号が小さくても同期しません。このようにして、NTP サーバのツリーは効率よく自律的に編成されています。

シスコの NTP 実装では、ストラタム 1 サービスをサポートしていないため、電波時計や原子時計に接続することはできません（ただし、いくつかの特定のプラットフォームでは、GPS 時刻源デバイスに接続できます）。ネットワークのタイム サービスは、IP インターネットで利用できる公開 NTP サーバから取得することを推奨します。

ネットワークがインターネットから切り離されている場合、シスコの NTP 実装では、実際には他の方法で時刻を決定している場合でも、NTP を介して同期されているものとして動作するようにマシンを設定できます。これにより、他のマシンが NTP を介してそのマシンと同期できるようになります。

自社のホスト システムに NTP ソフトウェアを組み込んでいるメーカーが数社あり、UNIX システム用のバージョンやその派生ソフトウェアも一般に入手できます。また、このソフトウェアにより UNIX 派生サーバは原子時計から時刻を直接取得することができ、シスコ ルータに時刻情報を伝えるようにすることもできます。

NTP を実行しているマシン間の通信（アソシエーション）は通常、静的に設定されており、各マシンには、アソシエーションを形成する必要があるすべてのマシンの IP アドレスが通知されます。アソシエーションが設定されたマシンの各ペアの間で NTP メッセージを交換することにより、正確な時刻管理が可能になります。

シスコの NTP 実装には、ネットワーク デバイスがネットワーク上で NTP 時刻情報を取得できる 3 つの方法があります。

- ホスト サーバへのポーリング
- NTP ブロードキャストのリスニング
- NTP マルチキャストのリスニング
- ホスト サーバへのポーリング
- NTP ブロードキャストのリスニング

LAN 環境では、IP ブロードキャスト メッセージまたは IP マルチキャスト メッセージを使用するように NTP を設定できます。ポーリングと比べ IP ブロードキャストまたは IP マルチキャスト メッセージではマシンごとにメッセージの送受信を設定するだけなので、複雑な設定作業が軽減されます。ただし、情報の流れが一方向に限定されるため、時刻管理の精度がわずかに低下します。

NTP ブロードキャスト クライアントは、指定した IPv4 アドレスにある NTP ブロードキャスト サーバから送信されるブロードキャスト メッセージをリスニングします。クライアントは最初に受信したブロードキャスト メッセージを使って、ローカルの時計を同期します。

NTP マルチキャスト サーバは、指定された IPv4 または IPv6 ローカル マルチキャスト グループ アドレスに定期的にメッセージを送信します。NTP マルチキャスト クライアントは、このアドレスで NTP メッセージをリスニングします。

マシン上の時刻は重要な情報であるため、NTP のセキュリティ機能を使用して、不正な時刻を誤って（または悪意を持って）設定できないように保護することを強く推奨します。その方法として、アクセス リストベースの制約方式と暗号化認証方式があります。

複数の時刻源（VINES、ハードウェア クロック、手動による設定）がある場合、NTP は常により信頼できる時刻源とされます。NTP の時刻は、他の方法による時刻に優先します。

GPS 週数ロールオーバー（WNRO）による問題の防止

- NTP ソース チェーンまたはサーバ チェーンに GPS ソースがない場合、GPS 週数ロールオーバー（WNRO）による影響はありません。
- GPS WNRO は、ユーザ トラフィックではなく、システム クロックにのみ影響を与えます。
- この条件に該当する GPS ソースを修正するには、GPS の製造元にお問い合わせください。

GPS WNRO の対象となる GPS ソースの影響を軽減するには、次のオプションの回避策を実行します。

- GPS ソースが 2019 年 4 月 6 日（またはそれ以降）に中断される可能性を引き起こす原因であると特定されている場合は、このソースに接続されているデバイスであるシスコとストラタム 1 デバイス上のクロックに ntp master を設定し、予防的措置として隔離します。この設定により、デバイスが下流の NTP クライアントへの同期用として独自のクロックを表示できるようになります。

（注） 前述のように、ntp master コマンドの使用は、この条件に対する回避策にすぎません。このコマンドは、ネットワーク全体への誤ったクロック値の分布を防止するために、GPS ソース関連の状況が解決されるまで使用します。

- 複数の NTP サーバ（3 つ以上、理想的には 4 つ）をネットワークのストラタム 2 レベルで構成し、ストラタム 2 レベルの NTP クライアントを有効にして、複数のストラタム 1 サーバからクロックを取得します。これにより、WNRO の影響を受けるストラタム 1 サーバは、WNRO の影響を受けないその他のストラタム 1 サーバと比較したときに、‘false ticker’ または ‘outlier’ クロック ソースとしてマークされるように調整されます。

NTP-PTP インターワーキング

NTP-PTP インターワーキングは、オペレーティング システムに時刻源として PTP、および Data over Cable Service Interface Specification（DOCSIS）Timing Interface（DTI）や Global Positioning System（GPS）などの他の有効な時刻（TOD）ソースを使用する機能を提供します。NTP-PTP

インターワーキングがサポートされる前は、オペレーティング システム時刻にはバックプレーンの時刻だけがサポートされていました。

NTP-PTP インターワーキングは、PTP および NTP プロセス間のステータス変更を通知する手段も提供します。また、起動、スイッチオーバー、またはカードおよびプロセス障害時に、オペレーティング システム時刻とバックプレーンの時刻の一義的な制御もサポートします。

Poll-Based アソシエーションの設定

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

ルータとその他のデバイス（ルータも可）間に、次のタイプの Poll-Based アソシエーションを設定できます。

- クライアント モード
- 対称アクティブ モード

クライアント モードと対称アクティブ モードは、高レベルの時刻の精度と信頼性を提供するために NTP が必要になる場合に使用します。

クライアント モードで動作しているネットワーキング デバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得します。次に、ネットワーキング デバイスは、ポーリングされたすべてのタイム サーバから、同期に使用するホストを選択します。この場合は、確立された関係がクライアントホスト関係なので、ホストがローカル クライアント デバイスから送信された時刻情報をキャプチャしたり使用したりすることはありません。このモードが最も適しているのは、他のローカル クライアントにどのような形式の時刻同期も提供する必要のない、ファイル サーバおよびワークステーションのクライアントです。ネットワーキング デバイスを同期させる時刻提供ホストを個別に指定し、クライアント モードで動作するようにネットワーキング デバイスを設定するには、 **server** コマンドを使用します。

対称アクティブ モードで動作しているネットワーキング デバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得し、そのホストによるポーリングに応答します。これはピアツーピアの関係であるため、ホストは通信相手のローカル ネットワーキング デバイスに関する時刻関連情報も保持します。相互に冗長な複数のサーバがダイバース ネットワーク パスを使用して相互に接続されている場合は、このモードを使用してください。現在のインターネットでは、ストラタム 1 サーバおよびストラタム 2 サーバのほとんどが、この形式のネットワーク設定を採用しています。ネットワーキング デバイスを同期させる時刻提供ホストを個別に指定し、対称アクティブ モードで動作するようにネットワーキング デバイスを設定するには、 **peer** コマンドを使用します。

他の複数のデバイスをポーリングして時刻を取得する場合、ルータは同期の対象となるデバイスを 1 台選択します。

（注） ルータと別のデバイス間のピアツーピア アソシエーションを設定するには、他のデバイスのピアとしてルータを設定する必要があります。

複数のピアおよびサーバを設定できますが、1 つの IP アドレスをピアとサーバの両方として同時に設定することはできません。

ピアからサーバ、またはサーバからピアへの特定の IP アドレスの設定を変更するには、 **peer** または **server** コマンドの **no** 形式を使用して、新しい設定を実行する前に現在の設定を削除します。新しい設定を実行する前に古い設定を削除しない場合、新しい設定によって古い設定は上書きされません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**server** *ip-address* [**vrf** *vrf*] [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**] [**burst**] [**iburst**] 例： `RP/0/RP0/CPU0:router(config-ntp)# server 172.16.22.44 minpoll 8 maxpoll 12`	他のシステムとのサーバアソシエーションを形成します。この手順を必要に応じて繰り返し、複数のデバイスとのアソシエーションを形成できます。
ステップ 4	**peer** *ip-address* [**vrf** *vrf*] [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**] 例： `RP/0/RP0/CPU0:router(config-ntp)# peer 192.168.22.33 minpoll 8 maxpoll 12 source tengige 0/0/0/1`	他のシステムとのピア アソシエーションを形成します。この手順を必要に応じて繰り返し、複数のシステムとのアソシエーションを形成できます。 （注） ルータとリモート デバイス間のピアツーピア アソシエーションの設定を完了するには、そのルータがリモート デバイス上でピアとして設定されている必要もあります。
ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

Broadcast-Based NTP アソシエーション

Broadcast-Based NTP アソシエーションでは、NTP サーバは、ネットワーク全体で NTP ブロードキャスト パケットを伝播します。ブロードキャスト クライアントは、NTP サーバによって伝搬されるブロードキャスト パケットをリッスンし、ポーリングには関与しません。

Broadcast-Based NTP アソシエーションは、時刻の精度および信頼性要件が緩やかであり、ネットワークがローカライズされ、クライアント数が多い（20 を超える）場合に使用します。また、帯域幅、システム メモリ、または CPU リソースが制限されているネットワークでも、

Broadcast-Based NTP アソシエーションの使用が推奨されます。情報の流れが一方向に限定されるため、Broadcast-Based NTP アソシエーションでは、時刻の精度がわずかに低下します。

ネットワークを通じて伝播される NTP ブロードキャスト パケットをリッスンするようにネットワーキング デバイスを設定するには、 **broadcast client** コマンドを使用します。ブロードキャスト クライアント モードが動作するためには、ブロードキャスト サーバとそのクライアントが同じサブネット上に存在する必要があります。 **broadcast** コマンドを使用して、NTP ブロードキャスト パケットを送信しているタイム サーバを特定のデバイスのインターフェイス上でイネーブルにする必要があります。

ネットワーキング デバイスが NTP ブロードキャスト パケットを送信するように設定するには、 **broadcast** コマンドを使用します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	（任意） **broadcastdelay** *microseconds* 例： `RP/0/RP0/CPU0:router(config-ntp)# broadcastdelay 5000`	NTP ブロードキャストの推定ラウンドトリップ遅延を調整します。
ステップ 4	**interface** *type interface-path-id* 例： `RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/1/0/0`	NTPインターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**broadcast client** 例： `RP/0/RP0/CPU0:router(config-ntp-int)# broadcast client`	指定されたインターフェイスが NTP ブロードキャスト パケットを受信するように設定します。 （注） インターフェイスが NTP ブロードキャスト パケットを送信するように設定するには、次の手順に移動します。

	コマンドまたはアクション	目的
ステップ 6	**broadcast** [**destination** *ip-address*] [**key** *key-id*] [**version** *number*] 例： `RP/0/RP0/CPU0:router(config-ntp-int)# broadcast destination 10.50.32.149`	指定されたインターフェイスが NTP ブロードキャスト パケットを送信するように設定します。 （注） インターフェイスが NTP ブロードキャスト パケットを受信するように設定するには、前の手順に移動します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp-int)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp-int)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

NTP アクセス グループの設定

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

アクセス リストベースの制限スキームを使用すると、ネットワーク全体、ネットワーク内のサブネット、またはサブネット内のホストに対し、特定のアクセス権限を許可または拒否できます。NTP 通信は、時刻要求と制御クエリーで構成されます。時刻要求とは、NTP サーバからの時刻同期の要求です。制御クエリーとは、NTP サーバからの設定情報の要求です。

アクセス グループのオプションは、次の順序で制限の緩いものから厳しいものへとスキャンされます。

1. **peer**：時刻要求と NTP 制御クエリーを許可し、システムがアクセス リストの基準を満たすアドレスを持つ別のシステムに同期することを許可します。
2. **serve**：時刻要求と NTP 制御クエリーを許可しますが、システムがアクセス リストの基準を満たすアドレスを持つ別のシステムに同期することは許可しません。
3. **serve-only**：アクセス リストの条件を満たすアドレスを持つシステムからの時刻要求のみを許可します。
4. **query-only**：アクセス リストの基準を満たすアドレスを持つ別のシステムからの NTP 制御クエリーのみを許可します。

複数のアクセス タイプについて送信元 IP アドレスがアクセス リストに一致する場合は、最初のタイプが認可されます。アクセス グループが指定されていない場合は、すべてのデバイスに対してすべてのアクセス タイプが認可されます。いずれかのアクセス グループが指定されている場合は、指定されたアクセス タイプだけが認可されます。

NTP 制御クエリーの詳細については、RFC 1305（NTP バージョン 3）を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** **例**： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**access-group** {**peer** | **query-only** | **serve** | **serve-only**} *access-list-name* **例**： `RP/0/RP0/CPU0:router(config-ntp)# access-group peer access1`	アクセス グループを作成して、基本的な IPv4 または IPv6 アクセス リストを適用します。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

NTP 認証の設定

ここでは、NTP 認証の設定方法について説明します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

信頼できる形式のアクセス コントロールが必要な場合は、暗号化された NTP 認証方式を使用する必要があります。IP アドレスに基づくアクセス リストベースの制約方式とは異なり、暗号化認証方式では認証キーと認証プロセスを使用して、ローカル ネットワーク上の指定されたピアまたはサーバによって送信された NTP 同期パケットが信頼できると見なすかどうかを、一緒に送信された時刻情報を受け入れる前に判断します。

認証プロセスは、NTP パケットが作成されるとすぐに開始されます。MD5 メッセージ ダイジェスト アルゴリズムを使用してメッセージ認証コード（MAC）が計算され、その MAC が NTP 同期パケットに埋め込まれます。NTP 同期パケットは、埋め込まれた MAC およびキー番号とともに受信側クライアントに送信されます。認証がイネーブルであり、キーが信頼できれば、受信側クライアントは同じ方法で MAC を計算します。計算された MAC と埋め込まれた MAC が一致すると、システムはパケットでこのキーを使用するサーバとの同期を許可されます。

NTP 認証が適切に設定されると、ネットワーキング デバイスは信頼できる時刻源と同期し、信頼できる時刻源だけに同期を提供します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** **例**： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**authenticate** **例**： `RP/0/RP0/CPU0:router(config-ntp)# authenticate`	NTP 認証機能をイネーブルにします。
ステップ 4	**authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name* **例**： `RP/0/RP0/CPU0:router(config-ntp)# authentication-key 42 md5 clear key1`	認証キーを定義します。 • 各キーにはキー番号、タイプ、値が設定されており、オプションで名前が設定されます。現在サポートされているキー タイプは **md5** だけです。
ステップ 5	**trusted-key** *key-number* **例**： `RP/0/RP0/CPU0:router(config-ntp)# trusted-key 42`	`Defines trusted authentication keys.` • キーが信頼できる場合、このルータは NTP パケットでこのキーを使用するシステムとのみ同期します。
ステップ 6	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

特定のインターフェイス上の NTP サービスのディセーブル化

NTPサービスは、デフォルトではすべてのインターフェイスでディセーブルになっています。

なんらかの NTP コマンドを入力すると、NTP がグローバルにイネーブルになります。特定のインターフェイス上の NTP をオフにすることによって、NTP パケットが特定のインターフェイス経由で受信されることを選択的に防止できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **no interface** *type interface-path-id*
	• **interface** *type interface-path-id* **disable**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# no interface pos 0/0/0/1`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/0/0/1 disable`
		指定されたインターフェイスでNTPサービスをディセーブルにします。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュ

	コマンドまたはアクション	目的
		レーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

NTP パケットの送信元 IP アドレスの設定

デフォルトでは、ルータが送信する NTP パケットの送信元 IP アドレスは、その NTP パケットが送信されたインターフェイスのアドレスです。別の送信元アドレスを設定するには、この手順を使用します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**source** *type interface-path-id* 例：	IP 送信元アドレスの取得元のインターフェイスを設定します。

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router(config-ntp)# source POS 0/0/0/1`	（注） このインターフェイスは、すべての宛先に送信されるすべてのパケットの送信元アドレスに使用されます。特定のアソシエーションに送信元アドレスを使用する場合は、Poll-Based アソシエーションの設定 （24 ページ） に示す **peer** コマンドまたは **server** コマンドで **source** キーワードを使用します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続

	コマンドまたはアクション	目的
		するには、**commit** コマンドを使用します。

正規の NTP サーバとしてのシステムの設定

システムが外部の時刻源に同期化されていない場合でも、ルータが正規の NTP サーバとして動作するように設定することができます。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**master** *stratum* 例： `RP/0/RP0/CPU0:router(config-ntp)# master 9`	`Makes the router an authoritative NTP server.` （注） **master** コマンドの使用には注意が必要です。このコマンドを使用すると、有効な時刻源が容易に上書きされてしまいます。低いストラタム番号を設定する際には、特に注意が必要です。**master** コマンドを使用して同じネットワーク内の複数のマシンを設定した場合は、それらのマシンの時刻が一致していないと、時刻管理が不安定になることがあります。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

NTP-PTP インターワーキングの設定

PTP を時刻源として NTP が使用するように設定するには、このタスクを使用します。

始める前に

NTP-PTP インターワーキングを設定するには、ルータで PTP がサポートされていてイネーブルにされている必要があります。PTPがイネーブルでない場合、設定をコミットしようとすると、次のようなエラー メッセージが表示されます。

```
RP/0/RP0/CPU0:router(config)# ntp master primary-reference-clock
RP/0/RP0/CPU0:router(config)# commit

% Failed to commit one or more configuration items. Please issue
'show configuration failed' from this session to view the errors

RP/0/RP0/CPU0:router(config)# show configuration failed
[:::]
ntp
 master primary-reference-clock
!!% 'ip-ntp' detected the 'fatal' condition 'PTP is not supported on this platform'
!
end
```

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**master primary-reference-clock** 例： `RP/0/RP0/CPU0:router(config-ntp)# master primary-reference-clock`	PTP を NTP の時刻源に指定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモー

	コマンドまたはアクション	目的
		ドに戻ります。変更はコミットされません。 • **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

ハードウェア クロックの更新

ハードウェア クロック（システム カレンダー）が搭載されたデバイスでは、ハードウェア クロックを、ソフトウェア クロックから定期的に更新されるように設定できます。ソフトウェア クロック（NTP を使用して設定）の時刻と日付がハードウェア クロックよりも正確であるため、これは、NTP を使用するデバイスで推奨されます。ハードウェア クロックの時刻設定は、時間の経過とともにわずかにずれる可能性があります。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。
ステップ 3	**update-calendar** 例： `RP/0/RP0/CPU0:router(config-ntp)# update-calendar`	ルータがシステム カレンダーをソフトウェア クロックから定期間隔で更新するように設定します。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

外部基準クロックのステータスの確認

ここでは、NTP コンポーネントのステータスの確認方法について説明します。

（注） コマンドは任意の順序で入力できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**show ntp associations** [**detail**] [**location** *node-id*] 例： `RP/0/RP0/CPU0:router# show ntp associations`	NTP アソシエーションのステータスを表示します。
ステップ 2	**show ntp status** [**location** *node-id*] 例： `RP/0/RP0/CPU0:router# show ntp status`	NTP のステータスを表示します。

例

次に、 **show ntp associations** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router# show ntp associations

Tue Oct  7 11:22:46.839 JST

      address         ref clock     st  when  poll reach  delay  offset    disp
*~192.168.128.5    10.81.254.131     2     1    64  377    7.98  -0.560   0.108
+~dead:beef::2 vrf testAA
                   171.68.10.80      3    20    64  377    6.00  -2.832   0.046
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured


RP/0/RP0/CPU0:router# show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
+~127.127.1.1      127.127.1.1       5     5  1024   37     0.0    0.00   438.3
*~172.19.69.1      172.24.114.33     3    13  1024    1     2.0   67.16     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

次に、 **show ntp status** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router# show ntp status

Tue Oct  7 11:22:54.023 JST

Clock is synchronized, stratum 3, reference is 192.168.128.5
nominal freq is 1000.0000 Hz, actual freq is 1000.2725 Hz, precision is 2**24
reference time is CC95463C.9B964367 (11:21:48.607 JST Tue Oct  7 2008)
clock offset is -1.738 msec, root delay is 186.050 msec
root dispersion is 53.86 msec, peer dispersion is 0.09 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.0002724105 s/s
system poll interval is 64, last update was 66 sec ago


RP/0/RP0/CPU0:router# show ntp status
```

```
Clock is synchronized, stratum 4, reference is 172.19.69.1
nominal freq is 1000.0000 Hz, actual freq is 999.9988 Hz, precision is 2**26
reference time is C54C131B.9EECF6CA (07:26:19.620 UTC Mon Nov 24 2008)
clock offset is 66.3685 msec, root delay is 7.80 msec
root dispersion is 950.04 msec, peer dispersion is 3.38 msec
```

NTP の実装の設定例

Poll-Based アソシエーションの設定：例

次に、ルータのシステム クロックが IP アドレス 192.168.22.33 のタイム サーバ ホストとのピア アソシエーションを形成し、IP アドレス 10.0.2.1 および 172.19.69.1 のタイム サーバ ホストによって同期されるように設定する、NTP の設定例を示します。

```
ntp
  server 10.0.2.1 minpoll 5 maxpoll 7
  peer 192.168.22.33

  server 172.19.69.1
```

Broadcast-Based アソシエーションの設定：例

次に、インターフェイス 0/11/0/0 が NTP ブロードキャスト パケットを受信するように設定し、NTP クライアントと NTP ブロードキャスト サーバ間の推定ラウンドトリップ遅延を 2 マイクロ秒に設定する、NTP クライアントの設定例を示します。

```
ntp
  interface tengige 0/11/0/0
    broadcast client
    exit
  broadcastdelay 2
```

次に、インターフェイス 0/11/0/2 がブロードキャスト サーバになるように設定する、NTP サーバの設定例を示します。

```
ntp
  interface tengige 0/11/0/2
    broadcast
```

Multicast-Based アソシエーションの設定：例

次に、10 ギガビット イーサネット インターフェイス 0/11/0/0 がマルチキャスト クライアントになり、デフォルトのマルチキャスト グループ（IPv4 アドレス 224.0.1.1）に参加するように設定する、NTP マルチキャスト クライアントの設定例を示します。

```
ntp interface TenGigE 0/11/0/0
  multicast client
```

次に、10 ギガビット イーサネット インターフェイス 0/11/0/0 がマルチキャスト サーバになるように設定する、NTP マルチキャスト サーバの設定例を示します。

```
ntp interface TenGigE 0/11/0/0
  multicast destination 224.0.1.1
```

NTP アクセス グループの設定：例

次に、以下のアクセス グループの制約事項が適用される NTP アクセス グループの設定例を示します。

- peer の制約事項は、peer-acl というアクセス リストの条件を満たす IP アドレスに適用されます。
- serve の制約事項は、serve-acl というアクセス リストの条件を満たす IP アドレスに適用されます。
- serve-only の制約事項は、serve-only-acl というアクセス リストの条件を満たす IP アドレスに適用されます。
- query-only の制約事項は、query-only-acl というアクセス リストの条件を満たす IP アドレスに適用されます。

```
ntp
  peer 10.1.1.1
  peer 10.1.1.1
  peer 10.2.2.2
  peer 10.3.3.3
  peer 10.4.4.4
  peer 10.5.5.5
  peer 10.6.6.6
  peer 10.7.7.7
  peer 10.8.8.8
  access-group peer peer-acl
  access-group serve serve-acl
  access-group serve-only serve-only-acl
  access-group query-only query-only-acl
  exit
ipv4 access-list peer-acl
  10 permit ip host 10.1.1.1 any
  20 permit ip host 10.8.8.8 any
  exit
ipv4 access-list serve-acl
  10 permit ip host 10.4.4.4 any
  20 permit ip host 10.5.5.5 any
  exit
ipv4 access-list query-only-acl
  10 permit ip host 10.2.2.2 any
  20 permit ip host 10.3.3.3 any
  exit
ipv4 access-list serve-only-acl
  10 permit ip host 10.6.6.6 any
  20 permit ip host 10.7.7.7 any
```

```
  exit
```

NTP 認証の設定：例

次に、NTP 認証の設定例を示します。この例では、次のように設定されます。

- NTP 認証がイネーブルになります。
- 2 つの認証キーが設定されます（キー 2 およびキー 3）。
- ルータは、ソフトウェア クロックが、認証キー 2 を使用する IP アドレス 10.3.32.154 のピアのクロックと（またはその逆に）同期することを許可するように設定されます。
- ルータは、ソフトウェア クロックが、認証キー 3 を使用する IP アドレス 10.32.154.145 のデバイスのクロックと同期することを許可するように設定されます。
- ルータは、NTP パケットに認証キー 3 を提供するシステムのみと同期するように設定されます。

```
ntp
  authenticate
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
```

インターフェイスでの NTP のディセーブル化：例

次に、0/11/0/0 インターフェイスをディセーブルにする NTP の設定例を示します。

```
ntp
  interface tengige 0/11/0/0
    disable
    exit
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  authenticate
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
```

NTP パケット用の送信元 IP アドレスの設定：例

次に、イーサネット管理インターフェイス 0/RP0/CPU0/0 が NTP パケットの送信元アドレスとして設定される、NTP の設定例を示します。

```
ntp
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
```

```
authentication-key 3 md5 encrypted 1311121E074110232621
authenticate
trusted-key 3
server 10.3.32.154 key 3
peer 10.32.154.145 key 2
source MgmtEth0/0/CPU0/0
```

正規の NTP サーバとしてのシステムの設定：例

次に、外部の NTP ソースが使用不可になったときに、独自の NTP マスター クロックを使用してピアと同期するように in which the router を設定する、NTP の設定例を示します。

```
ntp
 master 6
```

ハードウェア クロックの更新：例

次に、NTP の設定例を示します。 the router is configured to update its hardware clock from the software clock at periodic intervals:

```
ntp
 server 10.3.32.154
 update-calendar
```

VRF インターフェイス内での NTP サーバの設定

ここでは、VRF インターフェイス内に NTP サーバを設定する方法について説明します。

（注） 特定のコマンドで NTP をイネーブルにすることはできません。NTP は、最初に実行する NTP コンフィギュレーション コマンドによってイネーブルになります。

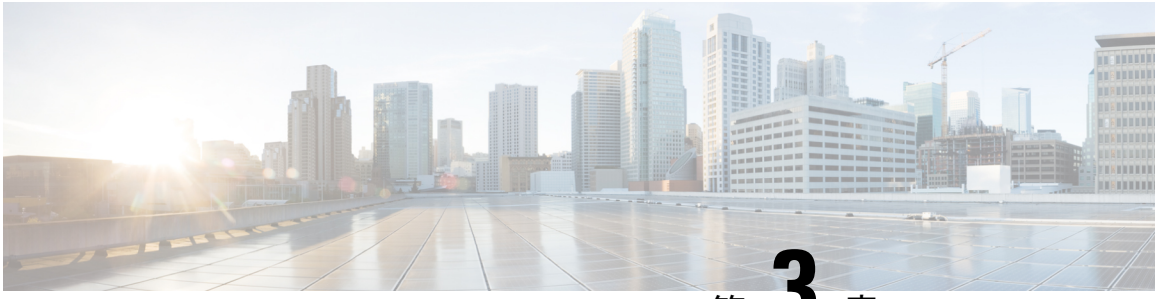
手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**ntp** 例： `RP/0/RP0/CPU0:router(config)# ntp`	NTP コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**vrf** *vrf-name* 例： `RP/0/RP0/CPU0:router(config)# ntp vrf Customer_A`	設定する VRF（VPN ルーティングおよび転送）インスタンスの名前を指定します。
ステップ 4	**source** *interface-type interface-instance* 例： `RP/0/RP0/CPU0:router(config)# ntp vrf Customer_A source bvi 70`	IP 送信元アドレスの取得元のインターフェイスを設定します。これにより、IOS-XR は VRF インターフェイス上の NTP クエリに応答できます。この場合、送信元は BVI です。 （注） このインターフェイスは、すべての宛先に送信されるすべてのパケットの送信元アドレスに使用されます。特定のアソシエーションに送信元アドレスを使用する場合は、Poll-Based アソシエーションの設定 （24 ページ） に示す **peer** コマンドまたは **server** コマンドで **source** キーワードを使用します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	または
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。



第 3 章

管理性の設定

このモジュールでは、Extensible Markup Language（XML）エージェント サービスをイネーブルにするために必要な設定について説明します。XML パーサー インフラストラクチャは、Document Object Model（DOM）、Simple アプリケーション プログラミング インターフェイス（API）for XML（SAX）、および文書型定義（DTD）の妥当性検査機能を使用した XML ドキュメントの解析と生成を実現します。

- DOM を使用すると、XML ドキュメントをプログラムによって作成、操作、生成できます。
- SAX は、XML タグ用のユーザ定義の関数をサポートします。
- DTD は、定義されたドキュメント タイプの妥当性検査を可能にします。

- XML の管理機能について （49 ページ）
- 管理機能の設定方法 （50 ページ）
- 管理機能の設定例 （51 ページ）

XML の管理機能について

Cisco IOS XR Extensible Markup Language（XML）API は、外部管理アプリケーションが使用するルータとのプログラマブル インターフェイスを実現します。このインターフェイスは、XML 形式の要求および応答ストリームを使用するルータ設定とモニタリングのメカニズムを提供します。XMLインターフェイスは、管理データ API（MDA）の上に構築されています。これは、Cisco IOS XR コンポーネントが、MDA スキーマ定義ファイルを介してデータ モデルをパブリッシュできるようにするメカニズムを提供します。

Cisco IOS XR ソフトウェア には、専用 TCP 接続、Secure Socket Layer（SSL）、または特定の VPN ルーティングおよび転送（VRF）インスタンスを使用して XML 経由でルータにアクセスする機能があります。

管理機能の設定方法

XML エージェントの設定

ここでは、XML エージェントの設定方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	**xml agent**［**ssl**］ 例： `RP/0/RP0/CPU0:router(config)# xml agent`	専用の TCP 接続を介して Extensible Markup Language（XML）要求をイネーブルにし、XML エージェント コンフィギュレーション モードを開始します。Secure Socket Layer（SSL）上での XML 要求をイネーブルにするには、**ssl** キーワードを使用します。
ステップ 2	**iteration on size** *iteration-size* 例： `RP/0/RP0/CPU0:router(config-xml-agent)# iteration on size 500`	大きい XML エージェントの応答の反復サイズを KB 単位で設定します。デフォルト値は 48 です。
ステップ 3	**session timeout** *timeout* 例： `RP/0/RP0/CPU0:router(config-xml-agent)# session timeout 5`	XML エージェントのアイドル タイムアウトを分単位で設定します。デフォルトでは、タイムアウトは設定されていません。
ステップ 4	**throttle** {**memory** *size* | **process-rate** *tags*} 例： `RP/0/RP0/CPU0:router(config-xml-agent)# throttle memory 300`	XML エージェントの処理能力を設定します。 • MB 単位でメモリ サイズを指定します。有効値の範囲は、100 ～ 600 です。IOS XR 64 ビットでは、値の範囲は 100 ～ 1024 です。デフォルトは 300 です。 • XML エージェントが 1 秒に処理できるタグ数の処理率を指定します。有効値の範囲は、1000 ～ 30000 です。デフォルトでは処理率は抑制されません。

	コマンドまたはアクション	目的
ステップ 5	**vrf** { **vrfname** | **default**} [**ipv4 access-list** *access-list-name*] 例： `RP/0/RP0/CPU0:router(config-xml-agent)# vrf vrf1`	指定された VPN ルーティングおよび転送（VRF）インスタンスでメッセージを送受信するように、専用エージェントまたは SSL エージェントを設定します。

管理機能の設定例

XML エージェントでの VRF のイネーブル化：例

次に、専用 XML エージェントを VRF1、VRF2 およびデフォルト VRF 経由でメッセージを送受信するように設定する例を示します。

```
RP/0/RP0/CPU0:router(config)# xml agent
RP/0/RP0/CPU0:router(config-xml-agent)# vrf VRF1
RP/0/RP0/CPU0:router(config-xml-agent)# vrf VRF2
```

次に、専用エージェントから VRF2 へのアクセスを削除する例を示します。

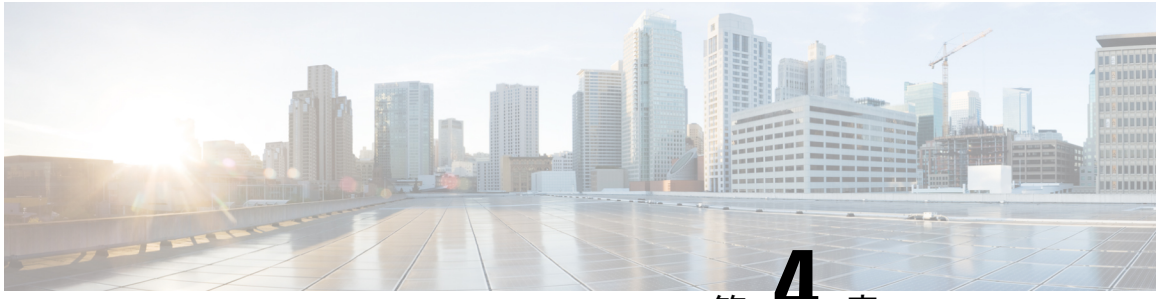
```
RP/0/RP0/CPU0:router(config)# xml agent ssl
RP/0/RP0/CPU0:router(config-xml-ssl)# vrf VRF1
RP/0/RP0/CPU0:router(config-xml-ssl-vrf)# vrf VRF2

RP/0/RP0/CPU0:router(config)# xml agent
RP/0/RP0/CPU0:router(config-xml-agent)# no vrf VRF1
```

次に、XML SSL エージェントを VRF1、VRF2 およびデフォルト VRF 経由でメッセージを送受信するように設定する例を示します。

```
RP/0/RP0/CPU0:router(config)# xml agent ssl
RP/0/RP0/CPU0:router(config-xml-agent)# vrf VRF1
RP/0/RP0/CPU0:router(config-xml-agent)# vrf VRF2
```

次に、専用 XML エージェントから VRF2 へのアクセスを削除する例を示します。

```
RP/0/RP0/CPU0:router(config)# xml agent ssl
RP/0/RP0/CPU0:router(config-xml-agent)# no vrf VRF2
```



第 4 章

オブジェクト トラッキングの設定

ここでは、Cisco IOS XR ネットワークでのオブジェクト トラッキングの設定について説明します。このモジュール内に記載されているコマンドの詳細については、「**その他の関連資料**」の項を参照してください。設定タスクを実行する手順の中で出現する可能性のあるその他のコマンドについて記載されたマニュアルを検索するには、トピック「その他の関連資料」の「**テクニカル ドキュメント**」の項を参照してください。

- オブジェクト トラッキングの設定 （53 ページ）
- オブジェクト トラッキングの実装の前提条件 （53 ページ）
- オブジェクト トラッキングについて （54 ページ）
- オブジェクト トラッキングの実装方法 （54 ページ）
- オブジェクト トラッキングの設定例 （65 ページ）

オブジェクト トラッキングの設定

ここでは、Cisco IOS XR ネットワークでのオブジェクト トラッキングの設定について説明します。このモジュール内に記載されているコマンドの詳細については、「**その他の関連資料**」の項を参照してください。設定タスクを実行する手順の中で出現する可能性のあるその他のコマンドについて記載されたマニュアルを検索するには、トピック「その他の関連資料」の「**テクニカル ドキュメント**」の項を参照してください。

オブジェクト トラッキングの実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

オブジェクト トラッキングについて

オブジェクト トラッキング とは、オブジェクトを追跡して、そのプロパティの変化に基づいて、トラッキング対象オブジェクトとは関係のない別のオブジェクトに対してアクションを実行する仕組みです。

各トラッキング対象オブジェクトは、トラッキング コマンドライン インターフェイス（CLI）で指定された一意の名前で識別されます。Cisco IOS XR が処理し、この名前を使用して特定のオブジェクトを追跡します。

トラッキング プロセスでは、定期的にトラッキング対象オブジェクトをポーリングして、ステートのアップ、ダウンなどの変化をユーザの指定により即時または時間をおいてレポートします。

リストを使った方法で複数のオブジェクトを追跡することもできます。リストはオブジェクトの組み合わせにブール論理式を使った柔軟なメソッドです。リストでは次の演算を使用します。

- **ブール AND 関数**：トラッキング対象リストにブール AND 関数を指定した場合、サブセット内に定義された各オブジェクトはアップ ステートでなければならないため、トラッキング対象オブジェクトもアップ ステートになります。
- **ブール OR 関数**：トラッキング対象リストにブール OR 関数を指定した場合、サブセット内に定義されたオブジェクトのうち少なくとも 1 つがアップ ステートでなければならないため、トラッキング対象オブジェクトもアップ ステートであることを意味します。

オブジェクト トラッキングの実装方法

ここでは、さまざまなオブジェクト トラッキングの手順を説明します。

インターフェイスのライン プロトコル ステートのトラッキング

インターフェイスのラインプロトコル ステートをトラッキングするには、グローバル コンフィギュレーション モードで次の作業を実行します。

インターフェイスのラインプロトコルがアップしている場合は、トラッキング対象オブジェクトはアップ状態と見なされます。

トラッキング対象オブジェクトの設定後、そのステートがトラッキング対象になっているインターフェイスを関連付けたり、トラッキング オブジェクトがインターフェイスをポーリングしてステートを取得するまで待機する秒数を指定したりすることができます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**track** *track-name* 例： `RP/0/RP0/CPU0:router(config)# track track1`	トラック コンフィギュレーション モードを開始します。 • *track-name*：トラッキングの対象となるオブジェクト名を指定します。
ステップ 3	**type line-protocol state** 例： `RP/0/RP0/CPU0:router(config-track)# type line-protocol state`	インターフェイスのライン プロトコルに基づいてトラッキングを作成します。
ステップ 4	**interface** *type interface-path-id* 例： `RP/0/RP0/CPU0:router(config-track-line-prot)# interface atm 0/2/0/0.1`	プロトコル ステートをトラッキングするインターフェイスを指定します。 • *type*：インターフェイス タイプを指定します。詳細については、疑問符（?）オンライン ヘルプ機能を使用します。 • *interface-path-id*：物理インターフェイスまたは仮想インターフェイスを識別します。 （注） ルータに現在設定されている可能性があるすべてのインターフェイスのリストを表示するには、 **show interfaces** コマンドを使用します。 （注） ループバック インターフェイスおよびヌル インターフェイスは、常にアップ ステートであり、そのためトラッキングできません。
ステップ 5	**exit** 例： `RP/0/RP0/CPU0:router(config-track-line-prot)# exit`	トラック ライン プロトコル コンフィギュレーション モードを終了します。
ステップ 6	（任意） **delay** {**up** *seconds* | **down** *seconds*} 例： `RP/0/RP0/CPU0:router(config-track)# delay up 10`	オブジェクトがアップかダウンかのトラッキング間に発生可能な遅延をスケジューリングします。

	コマンドまたはアクション	目的

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-track)# end`
	または
	`RP/0/RP0/CPU0:router(config-track)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IP ルートの到達可能性のトラッキング

ホストまたはネットワークがリモート サイトでダウン状態になった場合、ルーティング プロトコルはルータに通知し、ルーティング テーブルはそれに応じて更新されます。ルーティング プロセスは、ルーティング アップデートによってルートの状態が変わった場合にトラッキング プロセスに通知するように設定されます。

ルーティング テーブル エントリがルートに存在し、そのルートがアクセス可能であると、トラッキング対象オブジェクトはアップ状態にあると見なされます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**track** *track-name* 例： `RP/0/RP0/CPU0:router(config)# track track1`	トラック コンフィギュレーション モードを開始します。 • *track-name*：トラッキングの対象となるオブジェクト名を指定します。
ステップ 3	**type route reachability** 例： `RP/0/RP0/CPU0:router(config-track)# type route reachability vrf internet`	ルーティングアップデートによってルートの状態が変わった場合にトラッキングプロセスに通知するようにルーティングプロセスを設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **vrf** *vrf-table-name*
	• **route ipv4** *IP-prefix/mask*
	例：
	`RP/0/RP0/CPU0:router(config-track-route)# vrf vrf-table-4`
	または
	`RP/0/RP0/CPU0:router(config-track-route)# route ipv4 10.56.8.10/16`
		トラッキングする IP ルートのタイプを設定します。これは、ルータのタイプによって次のいずれかで構成可能です。
		• *vrf-table-name*：VRF テーブル名。
		• *IP-prefix/mask*：ネットワークとサブネット マスクからなる IP プレフィックス（例：10.56.8.10/16）。

ステップ 5	**exit** 例： `RP/0/RP0/CPU0:router(config-track-line-prot)# exit`	トラック ライン プロトコル コンフィギュレーション モードを終了します。
ステップ 6	（任意） **delay** {**up** *seconds* | **down** *seconds*} 例： `RP/0/RP0/CPU0:router(config-track)# delay up 10`	オブジェクトがアップかダウンかのトラッキング間に発生可能な遅延をスケジューリングします。
ステップ 7	**commit**	

オブジェクト リストに基づくトラッキングの設定

グローバル コンフィギュレーション モードでこのタスクを実行し、ブール式を使用してリストの状態を判断して、トラッキング対象オブジェクト リスト（ここではインターフェイスまたはプレフィックスのリスト）を作成します。

トラッキング対象リストには 1 つまたは複数のオブジェクトが含まれます。ブール式では、AND または OR 演算子を使用して 2 種類の演算を実行できます。たとえば、AND 演算子を使用して2つのインターフェイスをトラッキングする場合、アップは両方のインターフェイスがアップ状態であることを意味し、ダウンはいずれか一方のインターフェイスがダウン状態であることを意味します。

（注） トラッキング対象リストにオブジェクトを追加するには、そのオブジェクトが存在している必要があります。

NOT演算子は、1つまたは複数のオブジェクトに指定し、そのオブジェクトの状態を否定します。

トラッキング対象オブジェクトを設定したら、状態をトラッキングするインターフェイスを関連付ける必要があります。オプションとして、トラッキング オブジェクトがインターフェイスをポーリングしてその状態を取得するまでの待機時間を秒数で指定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**track** *track-name* **例**： `RP/0/RP0/CPU0:router(config)# track track1`	トラック コンフィギュレーション モードを開始します。 • *track-name*：トラッキングの対象となるオブジェクト名を指定します。
ステップ 3	**type list boolean** { **and** | **or** } **例**： `RP/0/RP0/CPU0:router(config-track-list)# type list boolean and`	ブール リスト オブジェクトを設定し、トラッキング リスト コンフィギュレーション モードを開始します。 • **boolean**：トラッキング対象リストのステートがブール式に基づいて決まることを指定します。 • **and**：リストについて、すべてのオブジェクトがアップの場合はアップ、ダウンのオブジェクトが1つ以上ある場合はダウンになるように指定します。たとえば2つのインターフェイスをトラッキングする場合、

	コマンドまたはアクション	目的
		アップは両方のインターフェイスがアップ状態であることを意味し、ダウンはいずれか一方のインターフェイスがダウン状態であることを意味します。 • **or**：少なくとも1つのオブジェクトがアップであればリストがアップになるように指定します。たとえば2つのインターフェイスをトラッキングする場合、アップはいずれか一方のインターフェイスがアップ状態であることを意味し、ダウンは両方のインターフェイスがダウン状態であることを意味します。
ステップ 4	**object** *object-name* ［ **not** ］ 例： `RP/0/RP0/CPU0:router(config-track-list)# object 3 not`	リストによるトラッキングの対象となるオブジェクトを指定します。 • *obect-name*：トラッキングするオブジェクトの名前。 • **not**：オブジェクトの状態を否定します。
ステップ 5	**exit** 例： `RP/0/RP0/CPU0:router(config-track-line-prot)# exit`	トラック ライン プロトコル コンフィギュレーション モードを終了します。
ステップ 6	（任意） **delay** {**up** *seconds* | **down** *seconds*} 例： `RP/0/RP0/CPU0:router(config-track)# delay up 10`	オブジェクトがアップかダウンかのトラッキング間に発生可能な遅延をスケジューリングします。
ステップ 7	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RP0/CPU0:router(config-track)# end` または	設定変更を保存します。 • **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router(config-track)# commit`	• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。 • **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。 • **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

オブジェクト リストに基づくトラッキングの設定：しきい値の割合

グローバル コンフィギュレーション モードでこのタスクを実行し、しきい値の割合を使用してリストの状態を判断して、トラッキング対象オブジェクト リスト（ここではインターフェイスまたはプレフィックスのリスト）を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**track** *track-name* 例： `RP/0/RP0/CPU0:router(config)# track track1`	トラック コンフィギュレーション モードを開始します。 • *track-name*：トラッキングの対象となるオブジェクト名を指定します。

	コマンドまたはアクション	目的
ステップ 3	**type list threshold percentage** 例： `RP/0/RP0/CPU0:router(config-track-list)# type list threshold percentage`	トラッキングのタイプにしきい値の割合リストを設定します。

ステップ 4	**object** *object-name*
	例：
	`RP/0/RP0/CPU0:router(config-track-list-threshold)# object 1`
	`RP/0/RP0/CPU0:router(config-track-list-threshold)# object 2`
	`RP/0/RP0/CPU0:router(config-track-list-threshold)# object 3`
	`RP/0/RP0/CPU0:router(config-track-list-threshold)# object 4`
		トラック タイプ track1 のメンバーに object 1、object 2、object 3 および object 4 を設定します。

ステップ 5	**threshold percentage up** *percentage* **down** *percentage* 例： `RP/0/RP0/CPU0:router(config-track-list-threshold)# threshold percentage up 50 down 33`	リストがそれぞれアップ状態またはダウン状態であると見なされるために、アップ状態またはダウン状態である必要があるオブジェクトの割合を設定します。 たとえば、object 1、object 2、および object 3 がアップ状態にあり、object 4 がダウン状態にある場合、リストはアップ状態にあると見なされます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-track)# end`
	または
	`RP/0/RP0/CPU0:router(config-track)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モー

	コマンドまたはアクション	目的
		ドに戻ります。変更はコミットされません。 • **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

オブジェクト リストに基づくトラッキングの設定：しきい値の重み

グローバル コンフィギュレーション モードでこのタスクを実行し、しきい値の重みを使用してリストの状態を判断して、トラッキング対象オブジェクト リスト（ここではインターフェイスまたはプレフィックスのリスト）を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**track** *track-name* 例： `RP/0/RP0/CPU0:router(config)# track track1`	トラック コンフィギュレーション モードを開始します。 • *track-name*：トラッキングの対象となるオブジェクト名を指定します。
ステップ 3	**type list threshold weight** 例： `RP/0/RP0/CPU0:router(config-track-list)# type list threshold weight`	トラッキングのタイプにしきい値の重みリストを設定します。

ステップ 4	**object** *object-name* **weight** *weight*
	例：
	`RP/0/RP0/CPU0:router(config-track-list-threshold)# object 1 weight 10`
	`RP/0/RP0/CPU0:router(config-track-list-threshold)#`
		track t1 のメンバーに object 1、object 2 および object 3 を設定し、それぞれに重み 10、5 および 3 を設定します。

	コマンドまたはアクション	目的
	`object 2 weight 5` `RP/0/RP0/CPU0:router(config-track-list-threshold)# object 3 weight 3`	
ステップ 5	**threshold weight up** *weight* **down** *weight* 例： `RP/0/RP0/CPU0:router(config-track-list-threshold)# threshold weight up 10 down 5`	リストがそれぞれアップ状態またはダウン状態であると見なされるために、アップ状態またはダウン状態である必要があるオブジェクトの重みの範囲を設定します。この例では、object 1 および 2 がアップ状態にあり、累積の重みは 15 である（10～5 の範囲内ではない）ため、リストはダウン状態と見なされます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RP0/CPU0:router(config-track)# end`
	または
	`RP/0/RP0/CPU0:router(config-track)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続

	コマンドまたはアクション	目的
		するには、**commit** コマンドを使用します。

IPSLA の到達可能性のトラッキング

IP サービス レベル契約（SLA）動作の戻りコードのトラッキングをイネーブルにするには、このタスクを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**track** *track-name* **例：** `RP/0/RP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	**type rtr** *ipsla-no* **reachability** **例：** `RP/0/RP0/CPU0:router(config-track)# type rtr 100 reachability`	到達可能性をトラッキングする IP SLA 動作 ID を指定します。*ipsla-no* の有効値は、1 ～ 2048 の範囲です。
ステップ 4	**commit**	

IPSLA トラッキングの設定：例

次に、IPSLA のトラッキング設定の例を示します。

```
RP/0/RP0/CPU0:router(config)# track track1
RP/0/RP0/CPU0:router(config-track)# type rtr 1 reachability
RP/0/RP0/CPU0:router(config-track)# delay up 5
RP/0/RP0/CPU0:router(config-track)# delay down 10
```

オブジェクト トラッキングの設定例

インターフェイスがアップ状態かダウン状態かのトラッキング：実行コンフィギュレーションの例

```
track connection100
  type list boolean and
    object object3 not
    delay up 10
    !
interface service-ipsec 23
  line-protocol track connection100
  !
```

インターフェイスのライン プロトコル ステートのトラッキング：実行コンフィギュレーションの例

この例では、トラフィックはインターフェイス service-ipsec1 から到着し、インターフェイス TenGigE0/11/0/3 を経由して終了します。

```
track IPSec1
  type line-protocol state
    interface TenGigE0/11/0/3
    !
interface service-ipsec 1
  ipv4 address 70.0.0.1 255.255.255.0
  profile vrf1_profile_ipsec
  line-protocol track IPSec1
  tunnel source 80.0.0.1
  tunnel destination 80.0.0.2
  service-location preferred-active 0/0/1
  !
```

次に、前述の例を実行した後の **show track** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router# show run track

  Track IPSec1
  Interface GigabitEthernet0_0_0_3 line-protocol
  !
    Line protocol is UP
    1 change, last change 10:37:32 UTC Thu Sep 20 2007
    Tracked by:
    service-ipsec1
    !
```

IP ルートの到達可能性のトラッキング：実行コンフィギュレーションの例

この例では、インターフェイス service-ipsec1 から到着したトラフィックの宛先がネットワーク 7.0.0.0/24 にあります。このトラッキング手順は、ルーティング プロトコル プレフィックスの状態に従い、ルーティング テーブルに変更があったときに信号を送ります。

```
track PREFIX1
  type route reachability
    route ipv4 7.0.0.0/24
    !
  interface service-ipsec 1
  vrf 1
  ipv4 address 70.0.0.2 255.255.255.0
  profile vrf_1_ipsec
  line-protocol track PREFIX1
  tunnel source 80.0.0.2
  tunnel destination 80.0.0.1
  service-location preferred-active 0/2/0
```

オブジェクトのリストに基づいたトラックの構築：実行コンフィギュレーションの例

この例では、インターフェイス service-ipsec1 から到着するトラフィックが、インターフェイス TenGigE0/11/0/3 およびインターフェイス ATM 0/2/0/0.1 を介して終了します。トラフィックの宛先はネットワーク 7.0.0.0/24 です。

いずれかのインターフェイスまたはリモート ネットワークがダウンした場合は、トラフィック フローが停止される必要があります。これを行うには、ブール AND 式を使用します。

```
track C1
 type route reachability
  route ipv4 3.3.3.3/32
 !
!
track C2
 type route reachability
  route ipv4 1.2.3.4/32
 !
!
track C3
 type route reachability
  route ipv4 10.0.20.2/32
 !
!
track C4
 type route reachability
  route ipv4 10.0.20.0/24
 !
!
track OBJ
 type list boolean and
  object C1
  object C2
 !
```

```
!
track OBJ2
 type list boolean or
  object C1
  object C2
 !
```

IPSLA ベースのオブジェクト トラッキングの設定：コンフィギュレーションの例

次に、ACL と IPSLA 設定を含む IPSLA ベースのオブジェクト トラッキングの設定例を示します。

ACL の設定：

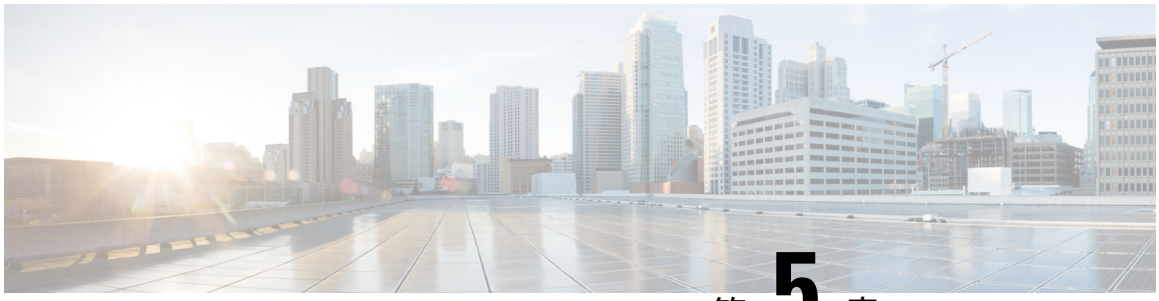
```
RP/0/RP0/CPU0:router(config)# ipv4 access-list abf-track
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit any nexthop track track1 1.2.3.4
```

オブジェクト トラッキングの設定：

```
RP/0/RP0/CPU0:router(config)# track track1
RP/0/RP0/CPU0:router(config-track)# type rtr 1 reachability
RP/0/RP0/CPU0:router(config-track)# delay up 5
RP/0/RP0/CPU0:router(config-track)# delay down 10
```

IPSLA の設定：

```
RP/0/RP0/CPU0:router(config)# ipsla
RP/0/RP0/CPU0:router(config-ipsla)# operation 1
RP/0/RP0/CPU0:router(config-ipsla-op)# type icmp echo
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo)# source address 2.3.4.5
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo)# destination address 1.2.3.4
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo)# frequency 60
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo)# exit
RP/0/RP0/CPU0:router(config-ipsla-op)# exit
RP/0/RP0/CPU0:router(config-ipsla)# schedule operation 1
RP/0/RP0/CPU0:router(config-ipsla-sched)# start-time now
RP/0/RP0/CPU0:router(config-ipsla-sched)# life forever
```



第 5 章

物理端末および仮想端末の設定

ライン テンプレートは、物理端末回線および仮想端末回線（VTY）を介した着信および送信転送の標準属性の設定を定義します。VTY プールを使用して、さまざまな仮想端末回線にテンプレートの設定を適用します。

ここでは、Cisco IOS XR ネットワークでの物理端末および仮想端末の実装に必要なタスクについて説明します。

- 物理端末と仮想端末を実装するための前提条件 （69 ページ）
- 物理端末および仮想端末の実装について （69 ページ）
- Cisco IOS XR ソフトウェアでの物理および仮想端末の実装方法 （72 ページ）
- 物理および仮想端末の実装の設定例 （77 ページ）

物理端末と仮想端末を実装するための前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

物理端末および仮想端末の実装について

物理端末および仮想端末を実装するには、この項の内容を理解しておく必要があります。

ライン テンプレート

Cisco IOS XR ソフトウェアでは次のライン テンプレートが使用できます。

- デフォルト ライン テンプレート：物理および仮想端末回線に適用されます。
- コンソール ライン テンプレート：コンソール回線に適用されます。
- ユーザ定義ライン テンプレート：仮想端末回線の範囲に適用できます。

ライン テンプレート コンフィギュレーション モード

ライン テンプレートの属性の変更は、ライン テンプレート コンフィギュレーション モードで行います。ラインテンプレート コンフィギュレーションモードに移行するには、XR コンフィギュレーション モードから **line** コマンドを実行し、変更するテンプレートを指定します。これらのライン テンプレートは、**line** コマンドを使用して設定できます。

- console：コンソール テンプレート
- default：デフォルト テンプレート
- template：ユーザ定義 テンプレート

line コマンドでテンプレートを指定すると、ルータは指定されたラインの端末属性を設定できるライン テンプレート コンフィギュレーション モードを開始します。次に、コンソールの属性を指定する例を示します。

```
RP/0/RP0/CPU0:router(config)# line console
RP/0/RP0/CPU0:router(config-line)#
```

ライン テンプレート コンフィギュレーション モードで、すべての使用可能なオプションを表示するには、オンラインヘルプ機能（**?**）を使用します。次に、便利なオプションの一部を示します。

- absolute-timeout：ライン切断のタイムアウト値を指定します。
- escape-character：ラインのエスケープ文字を変更します。
- exec-timeout：EXEC タイムアウトを指定します。
- length：画面に表示する行数を設定します。
- session-limit：許容される発信接続の数を指定します。
- session-timeout：入力トラフィックがない場合に接続を切断するインターバルを指定します。
- timestamp：各コマンドの前にタイムスタンプを表示します。
- width：表示端末の幅を指定します。

ライン テンプレート ガイドライン

コンソールテンプレートの変更およびユーザ定義テンプレートの設定について、次のガイドラインが適用されます。

- ルータ上の物理端末回線（コンソール ポート）のテンプレートは、ライン テンプレート コンフィギュレーション モードから変更します。コンソール テンプレートでライン テンプレート コンフィギュレーション モードを開始するには、XR コンフィギュレーション モードから **line console** コマンドを使用します。

- 仮想回線のテンプレートは、 **line** *template-name* コマンドでユーザ定義テンプレートを設定し、ライン テンプレート コンフィギュレーションからユーザ定義テンプレートの端末属性を設定し、 **vty pool** コマンドを使用して複数の仮想端末回線にテンプレートを適用することによって変更します。

（注） VTY プールを作成または変更する前に、XR コンフィギュレーション モードで **telnet server** コマンドを使用して telnet サーバをイネーブルにします。詳細については、『Cisco IOS XR IP Addresses and Services Configuration Guide』および『Cisco IOS XR IP Addresses and Services Command Reference』を参照してください。

端末の識別

コンソール ポート用の物理端末回線は、各コンソール ポートが存在するアクティブまたはスタンバイ ルート プロセッサ（RP）上での位置（*rack*/*slot*/*module* の形式で表される）によって識別されます。仮想端末の場合、物理的な位置は適用できません。Cisco IOS XR ソフトウェアは、VTY 接続が確立された順序に従って VTY ID を VTY に割り当てます。

VTY プール

各仮想ラインは、共通のライン テンプレート コンフィギュレーションを使用する接続プールのメンバーです。複数の VTY プールが存在する場合があり、それぞれ、VTY プールに設定されているとおりに、定義された数の VTY が含まれます。Cisco IOS XR ソフトウェアは、デフォルトで次の VTY プールをサポートします。

- デフォルトの VTY プール：デフォルトの VTY プールは、5 つの VTY（VTY 0 ～ 4）で構成され、それぞれデフォルト ライン テンプレートを参照します。
- デフォルトの障害マネージャ プール：デフォルトの障害マネージャ プールは、6 つの VTY（VTY 100 ～ 105）で構成され、それぞれデフォルト ライン テンプレートを参照します。

デフォルトの VTY プールおよびデフォルトの障害マネージャ プールのほかに、デフォルト テンプレートまたはユーザ定義テンプレートを参照できる、ユーザ定義の VTY プールを設定することもできます。

VTY プールを設定する際は、次のガイドラインに従ってください。

- デフォルト VTY プールの VTY の範囲は、VTY 0 から開始し、5 つ以上の VTY を含む必要があります。
- 0 ～ 99 の範囲の VTY は、デフォルトの VTY プールを参照できます。
- 5 ～ 99 の範囲の VTY は、ユーザ定義の VTY プールを参照できます。
- 100 以上の範囲の VTY は、障害マネージャの VTY プール用に予約されています。
- 障害マネージャ VTY プールの VTY の範囲は、VTY 100 から開始し、6 つ以上の VTY を含む必要があります。

- 1 つの VTY がメンバになることができる VTY プールは 1 つだけです。別のプールにすでに含まれる VTY を含めると、VTY プールの設定は失敗します。
- VTY プールを設定するときにアクティブな VTY プールからアクティブな VTY を削除しようとすると、その VTY プールの設定は失敗します。

Cisco IOS XR ソフトウェアでの物理および仮想端末の実装方法

テンプレートの変更

ここでは、コンソール ライン テンプレートとデフォルト ライン テンプレートの端末属性を変更する方法について説明します。設定した端末属性によって、指定したテンプレートのテンプレート設定が変更されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

ステップ 2	**line** {**console** | **default**}
	例：
	`RP/0/RP0/CPU0:router(config)# line console`
	または
	`RP/0/RP0/CPU0:router(config)# line default`
		指定された回線テンプレートの回線テンプレート コンフィギュレーション モードが開始されます。
		• **console** ：コンソール テンプレートのライン テンプレート コンフィギュレーション モードが開始されます。
		• **default** ：デフォルト ライン テンプレートのライン テンプレート コンフィギュレーション モードが開始されます。

ステップ 3	ライン テンプレート コンフィギュレーション モードでコマンドを使用して、特定のテンプレートの端末属性を設定します。	—
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例：	設定変更を保存します。 • **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

	コマンドまたはアクション	目的

	`RP/0/RP0/CPU0:router(config-line)# end`
	または
	`RP/0/RP0/CPU0:router(config-line)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		• **no** と入力すると、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを継続するには、 **commit** コマンドを使用します。

VTY プールの作成および変更

このタスクでは、VTY プールを作成および変更する方法について説明します。

VTY プールを参照するようにデフォルトのライン テンプレートを設定する場合は、ステップ 3 〜ステップ 5（**line template** および **exit** コマンド）を省略できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**telnet** {**ipv4** | **ipv6**} **server max-servers** *limit* 例： `RP/0/RP0/CPU0:router(config)# telnet ipv4 server max-servers 10`	許可できるTelnetサーバの数を指定します。最大で100台までのTelnetサーバを許可できます。 （注） デフォルトでは、Telnet サーバは許可されていません。Telnet サーバを使用できるようにするには、このコマンドを設定する必要があります。
ステップ 3	**line template** *template-name* 例： `RP/0/RP0/CPU0:router(config)# line template 1`	ユーザ定義のテンプレートのライン テンプレート コンフィギュレーション モードを開始します。
ステップ 4	ライン テンプレート コンフィギュレーション モードでコマンドを使用して、特定のライン テンプレートの端末属性設定を設定します。	—
ステップ 5	**exit** 例： `RP/0/RP0/CPU0:router(config-line)# exit`	ライン テンプレート コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。

ステップ 6	**vty-pool** {**default** | *pool-name* | **eem**} *first-vty last-vty* [**line-template** {**default** | *template-name*}]
	例：
	`RP/0/RP0/CPU0:router(config)#vty-pool default 0 5 line-template default`
	または
	`RP/0/RP0/CPU0:router(config)#vty-pool pool1 5 50 line-template template1`
	または
	`RP/0/RP0/CPU0:router#vty-pool eem 100 105 line-template template1`
	`RP/0/RP0/CPU0:router(config)#vty-pool default 0 5 line-template template1`
		VTY プールを作成または変更します。
		• **line-template** キーワードを使用してライン テンプレートを指定しないと、VTY プールがデフォルトのライン テンプレートになります。
		• **default** ：デフォルトの VTY プールを設定します。
		• デフォルトの VTY プールは、VTY 0 から開始し、5 つ以上の VTY（VTY 0 ～ 4）を含む必要があります。
		• デフォルトの VTY プールを構成する VTY の範囲を大きくすることによって、デフォルトの VTY プールのサイズを変更できます。

	コマンドまたはアクション	目的
		• *pool-name* ：ユーザ定義の VTY プールを作成します。 • ユーザ定義のプールは、少なくとも VTY 5 から開始する必要があります。ただし、デフォルトの VTY プールのサイズが変更されたかどうかにもよります。 • デフォルトの VTY プールの VTY の範囲のサイズが変更された場合、デフォルト ライン テンプレートがない最初の範囲の値を使用します。たとえば、デフォルトの VTY プールの VTY の範囲が 10 個の VTY が含まれるよう変更されている場合は（VTY 0 ～ 9）、ユーザ定義の VTY プールの範囲の値は VTY 10 から始まるようにします。 • **eem** ：Embedded Event Manager のプールを設定します。 • デフォルトの Embedded Event Manager の VTY プールは、VTY 100 から開始し、6 つ以上の VTY（VTY 100 ～ 105）を含む必要があります。 • **line-template** *template-name* ：ユーザ定義のテンプレートを参照する VTY プールを設定します。
ステップ 7	**commit**	

端末および端末セッションのモニタリング

このタスクでは、物理回線および端末回線に使用可能な **show** EXEC コマンドを使用して、端末と端末セッションをモニタする方法について説明します。

（注） コマンドは任意の順序で入力できます。

手順

	コマンドまたはアクション	目的
ステップ 1	（任意） **show line** [**aux location** *node-id* | **console location** *node-id* | **vty** *number*] 例： `RP/0/RP0/CPU0:router# show line`	端末回線の端末パラメータを表示します。 • **show line aux location** *node-id* EXEC コマンドを指定すると、補助回線の端末パラメータが表示されます。 • **show line console location** *node-id* EXEC コマンドを指定すると、コンソールの端末パラメータが表示されます。 • **location** *node-id* キーワードおよび引数については、それぞれの補助回線またはコンソールポートが存在するルート プロセッサ（RP）の場所を入力します。 • *node-id* 引数は、*rack*/*slot*/*module* の形式で入力します。 • **show line vty** *number* EXEC コマンドを指定すると、指定した VTY の端末パラメータが表示されます。
ステップ 2	（任意） **show terminal** 例： `RP/0/RP0/CPU0:router# show terminal`	現在の端末回線の端末属性設定を表示します。
ステップ 3	（任意） **show users** 例： `RP/0/RP0/CPU0:router# show users`	ルータのアクティブ回線に関する情報を表示します。

物理および仮想端末の実装の設定例

コンソール テンプレートの変更：例

この設定例は、コンソール ラインテンプレートの端末属性の設定を変更する方法を示します。

```
line console
  exec-timeout 0 0
  escape-character 0x5a
  session-limit 10
  disconnect-character 0x59
  session-timeout 100
  transport input telnet
  transport output telnet
```

この設定例では、次の端末属性がコンソール ライン テンプレートに適用されます。

- 端末セッションの EXEC タイムアウトは 0 分、0 秒に設定されます。EXEC タイムアウトを 0 分、0 秒に設定すると、EXEC タイムアウト機能がディセーブルになります。したがって、端末セッションの EXEC セッションがタイムアウトになることはありません。
- エスケープ文字は 0x5a の 16 進数値に設定されます（0x5a の 16 進数値は「Z」の文字に変換されます）。
- 発信端末セッションのセッション制限は、10 接続に設定されます。
- 切断文字は 0x59 の 16 進数値に設定されます（0x59 の 16 進文字は「Y」の文字に変換されます）。
- 発信端末セッションのセッション タイムアウトは 100 分（1 時間 40 分）に設定されます。
- 着信端末セッションに許可されるトランスポート プロトコルは、Telnet です。
- 発信端末セッションに許可されるトランスポート プロトコルは、Telnet です。

コンソール ラインテンプレートの端末属性がコンソールに適用されたことを確認するには、 **show line** コマンドを使用します。

```
RP/0/RP0/CPU0:router:router# show line console location 0/RP0/CPU0
Tue Nov 24 03:10:24.656 UTC
Tty             Speed     Overruns           Acc I/O
*con0/RP0/CPU0      9600       0/0               -/-

Line "con0_RP1_CPU0", Location "0/RP1/CPU0", Type "Console"
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600, "No" Parity, 2 stopbits, 8 databits
Template: console
Capabilities: Timestamp Enabled
Allowed transports are telnet.
```

デフォルト テンプレートの変更：例

次の設定例では、デフォルト ライン テンプレートの端末設定を上書きする例を示します。

```
line default
  exec-timeout 0 0
  width 512
  length 512
```

次の例では、次の端末属性はデフォルト ライン テンプレートのデフォルト端末属性の設定を上書きします。

- 端末セッションの EXEC タイムアウトは 0 分、0 秒に設定されます。EXEC タイムアウトを 0 分、0 秒に設定すると、EXEC タイムアウト機能をディセーブルにします。したがって、端末セッションの EXEC セッションは一切タイムアウトしません（デフォルト ライン テンプレートのデフォルトの EXEC タイムアウトは 10 分です）。
- デフォルト テンプレートを参照する端末の端末画面幅が 512 文字に設定されます（デフォルト ライン テンプレートのデフォルトの幅は 80 文字です）。
- デフォルト テンプレートを参照する端末に一度に表示する長さ、つまり行数は 512 行に設定されます（デフォルト ライン テンプレートのデフォルトの長さは 24 行です）。

デフォルト VTY プールを参照するユーザ定義テンプレートの設定：例

この設定例では、VTY 用のユーザ定義ライン テンプレート（この例では test という名前）を設定し、デフォルト VTY プールを参照するようにライン テンプレート テストを設定する方法を示します。

```
line template test
  exec-timeout 100 0
  width 100
  length 100
  exit
vty-pool default 0 4 line-template test
```

ユーザ定義の VTY プールを参照するユーザ定義テンプレートの設定：例

この設定例は、VTY のユーザ定義のライン テンプレート（この例では test2 という名前）を設定し、ユーザ定義の VTY プール（この例では pool1 という名前）を参照するライン テンプレート テストを設定をする方法を示します。

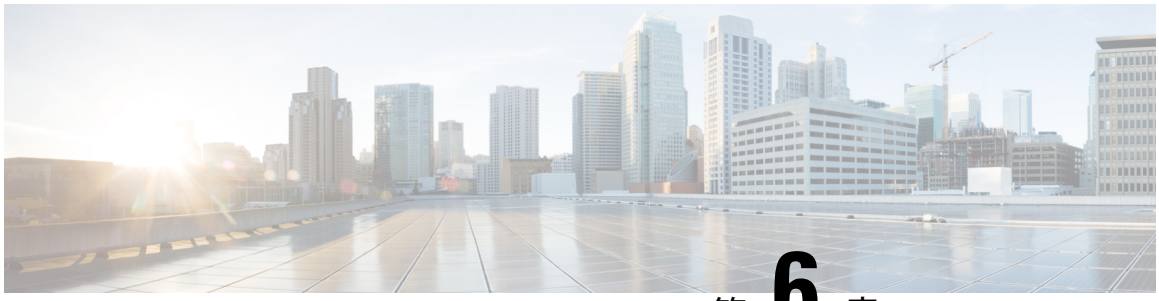
```
line template test2
  exec-timeout 0 0
  session-limit 10
  session-timeout 100
  transport input all
  transport output all
```

```
  exit
vty-pool pool1 5 50 line-template test2
```

障害マネージャの VTY プールを参照するユーザ定義テンプレートの設定：例

この設定例では、VTY のユーザ定義のライン テンプレート（この例では test3 という名前）を設定し、障害マネージャの VTY プールを参照するライン テンプレート テストを設定する方法を示します。

```
line template test3
  width 110
  length 100
  session-timeout 100
  exit
  vty-pool eem 100 106 line-template test3
```



第 6 章

簡易ネットワーク管理プロトコルの設定

簡易ネットワーク管理プロトコル（SNMP）は、アプリケーション層プロトコルであり、SNMP マネージャと SNMP エージェントとの通信に使用されるメッセージ フォーマットを提供します。SNMP では、ネットワーク内のデバイスのモニタリングと管理に使用する標準フレームワークと共通言語が提供されます。

ここでは、Cisco IOS XR ネットワーク上において SNMP の実装に必要な作業について説明します。

- SNMP の実装の前提条件 （81 ページ）
- Cisco IOS XR ソフトウェアでの SNMP の使用に関する制約事項 （81 ページ）
- SNMP の実装について （82 ページ）
- サブスクライバ セッションでのセッション MIB のサポート （89 ページ）
- Cisco IOS XR ソフトウェアでの SNMP の実装方法 （90 ページ）

SNMP の実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

Cisco IOS XR ソフトウェアでの SNMP の使用に関する制約事項

SNMP 出力は、32 ビット幅しかありません。そのため、2^{32} を超える情報は表示できません。2^{32} は 4.29 ギガビットになります。

（注） 10 ギガビット インターフェイスは 2^{32} を超えているため、インターフェイスに関する速度情報を表示しようとすると、結果が連結形式で表示される場合があります。

10 ギガビットを超えるインターフェイスの正しい速度を表示するには、ifHighSpeed を使用できます。

SNMP の実装について

SNMP を実装するには、この項の内容を理解しておく必要があります。

SNMP 機能の概要

SNMP フレームワークは 3 つの部分で構成されます。

- SNMP マネージャ
- SNMP エージェント
- 管理情報ベース（MIB）

SNMP マネージャ

SNMPマネージャは、SNMPを使用するネットワーク ホストのアクティビティを制御およびモニタするために使用されるシステムです。最も一般的な管理システムは、ネットワーク管理システム（NMS）と呼ばれます。NMS という用語は、ネットワーク管理に使用する専用デバイスを意味する場合と、このようなデバイス上で使用するアプリケーションを意味する場合があります。さまざまなネットワーク管理アプリケーションがSNMPとともに使用可能です。簡単なコマンドライン アプリケーションから機能が豊富なグラフィカル ユーザ インターフェイス（CiscoWorks 2000 製品ラインなど）まで、このような機能は多岐にわたっています。

SNMP エージェント

SNMPエージェントは、管理対象デバイスの内部で動作するソフトウェア コンポーネントであり、デバイスのデータを保持し、必要に応じて管理システムにそれらのデータを報告します。エージェントおよびMIB は、ルータに常駐します。SNMP エージェントをイネーブルにするには、マネージャとエージェントの関係を定義する必要があります。

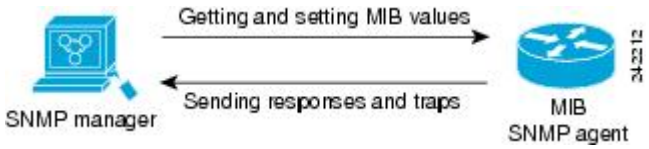
MIB

管理情報ベース（MIB）は、ネットワーク管理情報用の仮想情報ストレージ領域であり、管理対象オブジェクトの集合で構成されます。MIB 内には、MIB モジュールで定義された関連オブジェクトの集合体があります。MIB モジュールは、STD 58、RFC 2578、RFC 2579、および RFC 2580 の定義に従って、SNMP MIB モジュール言語で記述されます。なお、個々の MIB モジュールも MIB と呼ばれます。たとえば、インターフェイス グループ MIB（IF-MIB）はシステム上の MIB 内の MIB モジュールです。

SNMP エージェントには、SNMP マネージャが Get 操作や Set 操作を通じて値を要求したり変更したりできる MIB 変数が含まれています。マネージャでは、エージェントからの値の取得またはエージェントへの値の保存が可能です。エージェントは、デバイス パラメータやネットワーク データの保存場所であるMIBから値を収集します。エージェントは、マネージャのデータ取得要求やデータ設定要求にも応答できます。

次の図に、SNMPマネージャとSNMPエージェントの間の通信の関係を示します。マネージャは、MIB値の取得および設定の要求をエージェントに送信できます。エージェントはこれらの要求に応答できます。このやりとりとは別に、エージェント側からは、任意の通知（トラップ）をマネージャに送信して、ネットワークの状況をマネージャに通知できます。

図 *1 : SNMP* エージェントと *SMNP* マネージャの間の通信



IP-MIB のサポート

RFC4293 IP-MIB は、IPv4 と IPv6 の統計情報を個別に提供するように特別に設計されました。RFC 4293 で定義されている **ipIfStatsTable** には、インターフェイス固有の統計情報がリストされています。ipIfStatsTable の IPv6 統計情報のサポートは以前に追加されていますが、IP-MIB の IOS-XR 実装では、以前のリリースの場合 RFC4293 に従い IPv4 統計情報をサポートしていませんでした。

リリース 6.3.2 以降から、IP-MIB の IOS-XR 実装では、RFC4293 に従い IPv4 統計情報がサポートされています。これにより、インターフェイスごとに IPV4 と IPv6 の統計情報を個別に収集することができます。**ipIfStatsTable** は、2 つの**サブ ID アドレス タイプ（IPv4 または IPv6）**と**インターフェイス ifindex[1]** によってインデックス付けされます。IPv4 および IPv6 への IP-MIB サポートの実装は、読みやすさと保守性を向上させるためにリリース 6.3.2 から分離されています。

IPv4 統計情報について **ipIfStatsTable** に追加された OID のリストは次のとおりです。

- ipIfStatsInReceives
- ipIfStatsHCInReceives
- ipIfStatsInOctets
- ipIfStatsHCInOctets
- ipIfStatsOutTransmits
- ipIfStatsHCOutTransmits
- ipIfStatsOutOctets
- ipIfStatsHCOutOctets
- ipIfStatsDiscontinuityTime

iPv4 統計情報用に追加された新しい OID のリストについては、「SNMP OID Navigator」を参照してください。

SNMP バージョン

Cisco IOS XR ソフトウェア では、次のバージョンの SNMP がサポートされています。

- 簡易ネットワーク管理プロトコル バージョン 1（SNMPv1）
- 簡易ネットワーク管理プロトコル バージョン 2c（SNMPv2c）
- 簡易ネットワーク管理プロトコル バージョン 3（SNMPv3）

SNMPv1 および SNMPv2c はどちらも、コミュニティベース形式のセキュリティを使用します。エージェントの MIB にアクセスできるマネージャのコミュニティが、IP アドレス アクセス コントロール リストおよびパスワードによって定義されます。

SNMPv2c サポートには、バルク取得メカニズム、および管理ステーションに対するより詳細なエラー メッセージ報告が含まれています。バルク取得メカニズムは、テーブルおよび大量の情報の取得をサポートして、必要なラウンド トリップの回数を最小化します。SNMPv2c ではエラー処理のサポートが改善されました。たとえば、異なる種類のエラー条件が区別されるように、エラー コードが拡張されました。SNMPv1 では、これらの条件は単一のエラー コードを使用して報告されていました。エラー リターン コードでエラー タイプが報告されるようになりました。no such object exceptions、no such instance exceptions、および end of MIB view exceptions の 3 種類の例外も報告されます。

SNMPv3 は、セキュリティ モデルです。セキュリティ モデルは、ユーザおよびユーザが属するグループに合わせて設定される認証方式です。セキュリティ レベルとは、セキュリティ モデル内で許可されるセキュリティのレベルです。セキュリティ モデルとセキュリティ レベルの組み合わせによって、SNMP パケットの処理時に採用されるセキュリティ メカニズムが決まります。SNMPv3 で使用可能なセキュリティ レベルのリストについては、SNMPv1、SNMPv2、SNMPv3 のセキュリティ モデルおよびセキュリティ レベル （85 ページ） を参照してください。SNMPv3 機能は、RFC 3411 ～ 3418 をサポートします。

SNMP エージェントは、管理ステーションでサポートされる SNMP のバージョンを使用するように設定する必要があります。エージェントは複数のマネージャと通信できます。このため、1 つの管理ステーションとは SNMPv1 プロトコルを使用して通信し、1 つの管理ステーションとは SNMPv2c プロトコルを使用して通信し、もう 1 つの管理ステーションとは SMNPv3 を使用して通信することがサポートされるように、Cisco IOS-XR ソフトウェアを設定できます。

SNMPv1、SNMPv2c、および SNMPv3 の比較

SNMP v1、v2c、および v3 はすべて次の動作をサポートします。

- get-request：特定の変数から値を取得します。
- get-next-request：指定した変数の次の値を取得します。この動作はテーブル内からの変数取得によく使用されます。この動作では、SNMP マネージャに正確な変数名を認識させる必要はありません。SNMP マネージャは、必要な変数を MIB 内で順番に検索していきます。
- get-response：NMS によって送信された get-request、get-next-request、および set-request に応答する動作です。
- set-request：特定の変数に値を保存する動作です。

- trap：何らかのイベントが発生したときに、SNMP エージェントによって SNMP マネージャに送信される非送信請求メッセージです。

次の表では、SNMP v1、v2c、および v3 でサポートされるその他の主要な SNMP 機能を示します。

表 *1 : SNMPv1*、*v2c*、および *v3* 機能のサポート

機能	SNMP v1	SNMP v2c	SNMP v3
Get-Bulk 動作	×	○	○
Inform 動作	×	○（Cisco IOS XR ソフトウェアでは ×）	○（Cisco IOS XR ソフトウェアでは ×）
64 ビット カウンタ	×	○	○
テキストの表記法	×	○	○
認証	×	×	○
プライバシー（暗号化）	×	×	○
認証およびアクセス コントロール（ビュー）	×	×	○

SNMPv1、SNMPv2、SNMPv3 のセキュリティ モデルおよびセキュリティ レベル

セキュリティ レベルは、SNMP メッセージを開示から保護する必要があるかどうか、およびメッセージを認証するかどうか判断します。セキュリティ モデル内のさまざまなセキュリティ レベルは、次のとおりです。

- noAuthNoPriv：認証または暗号化を実行しないセキュリティ レベル。
- authNoPriv：認証は実行するが、暗号化を実行しないセキュリティ レベル。
- authPriv：認証と暗号化両方を実行するセキュリティ レベル。

SNMPv1、SNMPv2c、および SNMPv3 の 3 つのセキュリティ モデルを使用できます。セキュリティ モデルとセキュリティ レベルの組み合わせにより、SNMP メッセージの処理中に適用されるセキュリティ メカニズムが決まります。

次の表で、セキュリティ モデルとセキュリティ レベルの組み合わせについて説明します。

表 *2 : SNMP* セキュリティ モデルおよびセキュリティ レベル

モデル	レベル	認証	暗号化	結果
v1	noAuthNoPriv	コミュニティ ストリング	未対応	コミュニティ ストリングの照合を使用して認証します。

モデル	レベル	認証	暗号化	結果
v2c	noAuthNoPriv	コミュニティ ストリング	未対応	コミュニティストリングの照合を使用して認証します。
v3	noAuthNoPriv	ユーザ名	未対応	ユーザ名の照合を使用して認証します。
v3	authNoPriv	HMAC-MD5 または HMAC-SHA	非対応	HMAC[1]-MD5 [2] アルゴリズムまたは HMAC-SHA [3]に基づいて認証します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	DES	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。DES[4] 56 ビット暗号化、および CBC[5] DES（DES-56）標準に基づいた認証を提供します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	3DES	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。168 ビットの 3DES[6] レベルの暗号化を提供します。
v3	authPriv	HMAC-MD5 または HMAC-SHA	AES	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。128 ビットの AES[7] レベルの暗号化を提供します。

1 Hash-Based Message Authentication Code
2 メッセージ ダイジェスト 5
3 セキュア ハッシュ アルゴリズム
4 データ暗号規格
5 暗号ブロック連鎖
6 Triple Data Encryption Standard
7 Advanced Encryption Standard

3DESおよびAES暗号化規格を使用するため、セキュリティ パッケージ（k9sec）がインストールされている必要があります。ソフトウェア パッケージのインストールの詳細については、『*Upgrading and Managing Cisco IOS XR Software*』を参照してください。

SNMPv3 の利点

SNMPv3は、認証、暗号化、およびアクセス コントロールを提供することで、デバイスへの安全なアクセスを実現します。これらのセキュリティの利点が追加されたことより、次のセキュリティ上の脅威に対して SNMP がセキュリティ保護されます。

- マスカレード：SNMP ユーザが別の SNMP ユーザのアイデンティティを装って、その SNMP ユーザが許可されていない管理操作を実行する脅威。

- メッセージストリームの改変：メッセージが悪意を持って並べ替え、遅延、または再生されて（サブネットワーク サービスの通常の操作によって発生するよりも大きい程度に）、SNMP が不正な管理操作を実行するようになる脅威。
- 暴露：SNMP エンジン間でのやり取りが傍受される可能性がある脅威。ローカル ポリシーの問題としてこの脅威から保護が必要な場合があります。

さらに、SNMPv3 では、SNMP 管理対象オブジェクト上のプロトコル操作に対するアクセス制御も提供されます。

SNMPv3 のコスト

SNMPv3 の認証および暗号化は、MIB オブジェクトに対する SNMP 操作の実行時の応答時間をわずかに増加させる要因となります。このコストは、SNMPv3 がもたらすセキュリティ上の利点からすれば、無視できる程度のものです。

次の表に、セキュリティ モデルとセキュリティ レベルのさまざまな組み合わせを応答時間の短い順に示します。

表 *3 :* 応答時間の短い順

セキュリティ モデル	セキュリティ レベル
SNMPv2c	noAuthNoPriv
SNMPv3	noAuthNoPriv
SNMPv3	authNoPriv
SNMPv3	authPriv

ユーザベースのセキュリティ モデル

SNMPv3 ユーザベース セキュリティ モデル（USM）は SNMP メッセージレベル セキュリティを参照し、次のサービスを提供します。

- メッセージの完全性：メッセージが不正な方法で変更または破壊されず、データ シーケンスが悪意なく起こり得る範囲を超えて変更されていないことを保証します。
- メッセージ発信元の認証：受信データを発信したユーザのアイデンティティが確認されたことを保証します。
- メッセージの機密性：情報が使用不可であること、または不正なユーザ、エンティティ、またはプロセスに開示されないことを保証します。

SNMPv3 は、設定済みユーザによる管理動作のみを許可し、SNMP メッセージを暗号化します。

USM では、次の 2 つの認証プロトコルが使用されます。

- HMAC-MD5-96 認証プロトコル
- HMAC-SHA-96 認証プロトコル

USM は、メッセージ暗号化用のプライバシー プロトコルとして暗号ブロック連鎖（CBC）-DES（DES-56）を使用します。

View-Based Access Control Model

SNMP ユーザは、View-Based Access Control Model（VACM）を使用して、SNMP オブジェクトに対する読み取りアクセス、書き込みアクセス、または通知アクセスを指定することにより、SNMP 管理対象オブジェクトへのアクセスを制御できます。これは、ビューによって制限されているオブジェクトへのアクセスを防止します。これらのアクセス ポリシーは、 **snmp-server group** コマンドでユーザ グループを構成するときに設定できます。

MIB ビュー

セキュリティ上の理由から、一部のグループのアクセスを、管理ドメイン内の一部の管理情報のみに限定できることが頻繁に重要になります。この機能を実現するために、管理オブジェクトへのアクセスは、MIB ビューによって制御されます。このビューには、表示可能な管理対象オブジェクト タイプ（およびオプションとしてオブジェクト タイプの特定のインスタンス）のセットが含まれます。

アクセス ポリシー

アクセスポリシーによって、グループのアクセス権限が決定します。アクセス権限には、次の 3 種類があります。

- 読み取りビュー アクセス：オブジェクト読み取り時に、グループに許可されているオブジェクト インスタンスのセット。
- 書き込みビュー アクセス：オブジェクト書き込み時に、グループに許可されているオブジェクト インスタンスのセット。
- 通知ビューアクセス：オブジェクトの通知での送信時に、グループに許可されているオブジェクト インスタンスのセット。

SNMP の IP precedence および DSCP サポート

SNMP による IP precedence および差分化サービスコードポイント（DSCP; DiffServ コード ポイント）のサポートでは、SNMP トラフィックに特定した QoS を提供します。ユーザがプライオリティの設定を変更することができるため、ルータで生成した SNMP トラフィックを特定の QoS クラスに割り当てます。IP precedence または IP DSCP のコード ポイント値は、パケットを重み付けランダム早期検出（WRED）でどのように処理するかを決定するのに使用します。

ルータで生成された SNMP トラフィックに IP precedence または IP DSCP が設定されると、同じルータの種類の異なる SNMP トラフィックに異なる QoS クラスを割り当てられなくなります。

IP precedence 値は、IP ヘッダーの ToS（タイプ オブ サービス）バイトの最初の 3 ビットです。IP DSCP コード ポイント値は、差分化サービス（DiffServ フィールド）バイトの最初の 6 ビットです。最大 8 つの異なる IP precedence マーキングまたは 64 の異なる IP DSCP マーキングを設定できます。

サブスクライバセッションでのセッションMIBのサポート

SNMP モニタリングでは、すべてのタイプのサブスクライバに関する情報が必要です。CISCO-SUBSCRIBER-SESSION-MIB は、サブスクライバごとのデータと集約サブスクライバ（PPPoE）データをモデル化するために定義されます。設定されたしきい値を超える集約セッション数に関する通知（トラップ）をサポートする必要があります。
CISCO-SUBSCRIBER-SESSION-MIB の汎用 MIB データ コレクタ マネージャ（DCM）のサポートにより、データ収集が高速化し、並列データの処理も向上します。

SNMP 通知

SNMP の重要な機能の 1 つは、SNMP エージェントから通知を生成できることです。これらの通知では、要求を SNMP マネージャから送信する必要はありません。Cisco IOS XR ソフトウェアでは、任意（非同期）の通知は、トラップとしてのみ生成できます。トラップは、ネットワーク上のある状態を SNMP マネージャに通知するメッセージです。通知は、不正なユーザ認証、再起動、接続の切断、隣接ルータとの接続の切断、その他の重要なイベントを表示します。

（注） インフォーム要求（インフォーム操作）はCisco IOS XR ソフトウェア ではサポートされています。

トラップの信頼性はインフォームより低くなります。受信側はトラップを受信しても確認応答を送信しないからです。送信側は、トラップが受信されたかどうかを判断できません。インフォーム要求を受信する SNMP マネージャは、SNMP 応答プロトコルデータ ユニット（PDU）でメッセージの受信を確認応答します。マネージャがインフォーム要求を受信しなかった場合、応答は返されません。送信側が応答を受信しない場合、インフォーム要求を再び送信できます。このため、インフォームの方が目的の宛先に到達する確実性が高くなります。

ただし、インフォームはルータやネットワークのリソースをより多く消費するので、多くの場合、トラップの方が好んで使用されます。送信と同時に廃棄されるトラップと異なり、インフォーム要求は応答を受信するまで、または要求がタイムアウトになるまで、メモリ内に保持する必要があります。またトラップが一度だけ送信されるのに対し、インフォームは数回再試行されることがあります。再送信の回数が増えるとトラフィックが増加し、ネットワークのオーバーヘッドが高くなる原因にもなります。このように、トラップとインフォーム要求の間には、信頼性とリソースのトレードオフの関係があります。

図 *2 : SNMP* マネージャで受信したトラップ

この図では、エージェント ルータは SNMP マネージャにトラップを送信します。マネージャはトラップを受信しますが、エージェントに確認応答を送信しません。エージェントには、トラップが宛先に到達したことを知る方法がありません。

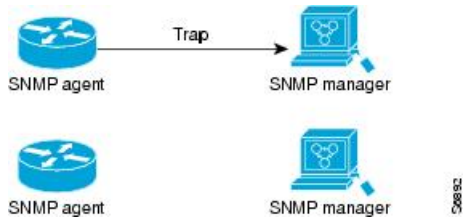
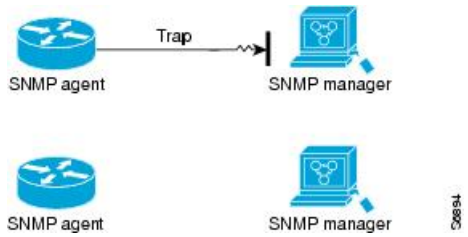


図 *3 : SNMP* マネージャで受信されなかったトラップ

次の図では、エージェントがマネージャにトラップを送信しますが、トラップはマネージャに届きません。トラップが宛先に到達しなかったことをエージェントが確認する方法がないため、トラップは再度送信されません。そのため、マネージャはこのトラップを受信できません。



セッション タイプ

サポートされているセッション タイプは次のとおりです。

- PPPoE
- IP SUB PKT
- IP SUB DHCP

Cisco IOS XR ソフトウェアでの SNMP の実装方法

ここでは、SNMP の実装方法について説明します。

snmp-server コマンドは、デフォルトで、管理イーサネット インターフェイスで SNMP をイネーブルにします。

SNMPv3 の設定

このタスクでは、ネットワーク管理およびモニタリングに SNMPv3 を設定する方法について説明します。

（注） 特定のコマンドでSNMPv3をイネーブルにすることはできません。SNMPv3は、最初に実行する **snmp-server** グローバル コンフィギュレーション コマンド（config）によってイネーブルになります。したがって、このタスクでは **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-server engineid local** *engine-id* 例： `RP/0/RP0/CPU0:router# snmp-server engineID local 00:00:00:09:00:00:00:a1:61:6c:20:61`	ローカル SNMP エンジンの識別番号を指定します。
ステップ 3	（任意） **snmp-server vrf** *vrf-name* 例： `RP/0/RP0/CPU0:router# snmp-server vrf vrfa`	SNMP の VRF プロパティを設定します。
ステップ 4	**snmp-server view** *view-name oid-tree* {**included** | **excluded**} 例： `RP/0/RP0/CPU0:router# snmp-server view view_name 1.3.6.1.2.1.1.5 included`	ビュー レコードを作成または変更します。
ステップ 5	**snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 write view_name2`	新規 SNMP グループ、または SNMP ユーザを SNMP ビューにマッピングするテーブルを設定します。
ステップ 6	**snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56**	SNMP グループに新しいユーザを設定します。

	コマンドまたはアクション	目的
	{**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server user noauthuser group_name v3`	
ステップ 7	**commit**	
ステップ 8	（任意） **show snmp** 例： `RP/0/RP0/CPU0:router# show snmp`	SNMP のステータスに関する情報を表示します。
ステップ 9	（任意） **show snmp engineid** 例： `RP/0/RP0/CPU0:router# show snmp engineid`	ローカル SNMP エンジンに関する情報を表示します。
ステップ 10	（任意） **show snmp group** 例： `RP/0/RP0/CPU0:router# show snmp group`	ネットワークの各 SNMP グループに関する情報を表示します。
ステップ 11	（任意） **show snmp users** 例： `RP/0/RP0/CPU0:router# show snmp users`	SNMP ユーザ テーブルの各 SNMP ユーザ名に関する情報を表示します。
ステップ 12	（任意） **show snmp view** 例： `RP/0/RP0/CPU0:router# show snmp view`	関連する MIB ビュー ファミリー名、ストレージタイプ、ステータスなど、設定されたビューに関する情報を表示します。

SNMPv3 の設定：例

エンジン ID の設定

次に、ローカル SNMP エンジンの ID を設定する例を示します。

```
snmp-server engineID local 00:00:00:09:00:00:00:a1:61:6c:20:61
```

（注） エンジン ID が設定されると、SNMP エージェントが再起動します。

ローカル SNMP エンジンの ID の確認

次に、ローカル SNMP エンジンの ID を確認する例を示します。

```
config
  show snmp engineid

  SNMP engineID 00000009000000a1ffffffff
```

ビューの作成

ビューを作成するには 2 つの方法があります。

- **snmp-server view** コマンドの **included** キーワードを使用することによって、ビューに MIB ファミリの ASN.1 サブツリーのオブジェクト識別子（OID）を包含することができます。
- **snmp-server view** コマンドの **excluded** キーワードを使用することによって、ビューから MIB ファミリの ASN.1 サブツリーの OID サブツリーを除外することができます。

次に、sysName（1.3.6.1.2.1.1.5）オブジェクトを含むビューを作成する例を示します。

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1.5 included
```

次に、システム グループのすべての OID を含むビューを作成する例を示します。

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
```

次に、除外されている sysName オブジェクト（1.3.6.1.2.1.1.5）を除く、システム グループのすべての OID を含むビューを作成する例を示します。

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1.5 excluded
```

設定したビューの確認

次に、設定したビューの情報を表示する例を示します。

```
RP/0/RP0/CPU0:router# show snmp view

  v1default 1.3.6.1 - included nonVolatile active
  SNMP_VIEW1 1.3.6.1.2.1.1 - included nonVolatile active
  SNMP_VIEW1 1.3.6.1.2.1.1.5 - excluded nonVolatile active
```

グループの作成

通知、読み取り、または書き込みビューを明示的に指定しないと、Cisco IOS XR ソフトウェアではv1デフォルト（1.3.6.1）が使用されます。次に、デフォルトビューを使用するグループを作成する例を示します。

```
RP/0/RP0/CPU0:router# snmp-server group group-name v3 auth
```

次の設定例は、グループに適用されるビューから除外された sysUpTime オブジェクト（1.3.6.1.2.1.1.3）を除く、システム内のすべての OID に対する読み取りアクセス権があり、sysName オブジェクト（1.3.6.1.2.1.1.5）に対しては書き込みアクセス権しかないグループを作成する例を示します。

```
!
  snmp-server view view_name1 1.3.6.1.2.1.1 included
  snmp-server view view_name1 1.3.6.1.2.1.1.3 excluded
  snmp-server view view_name2 1.3.6.1.2.1.1.5 included
  snmp-server group group_name1 v3 auth read view_name1 write view_name2
  !
```

グループの確認

この例では、設定したグループの属性を確認する方法を示します。

```
RP/0/RP0/CPU0:router# show snmp group

  groupname: group_name1                  security model:usm
  readview : view_name1                   writeview: view_name2
  notifyview: v1default
  row status: nonVolatile
```

ユーザの作成および確認

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp-server view view_name 1.3.6.1.2.1.1 included
  snmp-server group group_name v3 noauth read view_name write view-name
  !
```

次に、システム グループに対する読み取りビュー アクセスおよび書き込みビュー アクセスの権限を持つ noAuthNoPriv ユーザを作成する例を示します。

```
config
  snmp-server user noauthuser group_name v3
```

（注） noAuthNoPriv ユーザを作成するには、ユーザが noauth グループに属している必要があります。

次に、SNMP ユーザに適用する属性を確認する例を示します。

```
RP/0/RP0/CPU0:router# show snmp user

  User name: noauthuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server group SNMP_GROUP1 v3 auth notify SNMP_VIEW1 read SNMP_VIEW1 write SNMP_VIEW1

 !
```

次に、システム グループに対する認証（暗号化を含む）、読み取り/書き込みビュー アクセスの権限を持つユーザを作成する例を示します。

```
config
  snmp-server user userv3authpriv SNMP_GROUP1 v3 auth md5 password123 priv aes 128
password123
```

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp-server view view_name 1.3.6.1.2.1.1 included
  snmp group group_name v3 priv read view_name write view_name
  !
```

次に、システム グループに対する読み取りビュー アクセスおよび書き込みビュー アクセスの権限を持つ authNoPriv ユーザを作成する例を示します。

```
RP/0/RP0/CPU0:router# snmp-server user authuser group_name v3 auth md5 clear auth_passwd
```

（注） グループはセキュリティ レベル Auth に設定されているので、このグループにアクセスするには、ユーザが最低でも「auth」として設定されている必要があります（「priv」ユーザもこのグループにアクセスできます）。このグループに設定された authNoPriv ユーザの authuser は、ビューにアクセスするために認証パスワードを入力する必要があります。この例では、auth_passwd が認証パスワード文字列として設定されています。auth_passwd パスワード文字列の前に **clear** キーワードが指定されていることに注意してください。 **clear** キーワードは、指定されているパスワード文字列が暗号化されていないことを示します。

次に、SNMP ユーザに適用する属性を確認する例を示します。

```
RP/0/RP0/CPU0:router# show snmp user

  User name: authuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

次の SNMPv3 ビューおよび SNMPv3 グループの設定があるとします。

```
!
  snmp view view_name 1.3.6.1.2.1.1 included
  snmp group group_name v3 priv read view_name write view_name
  !
```

次に、システム グループへの読み取りビュー アクセスおよび書き込みビュー アクセスの権限を持つ authPriv ユーザを作成する例を示します。

```
config
  snmp-server user privuser group_name v3 auth md5 clear auth_passwd priv des56 clear
priv_passwd
```

（注） グループのセキュリティ レベルは Priv なので、ユーザがこのグループにアクセスするには、「priv」ユーザとして設定される必要があります。この例のユーザ privuser は、ビュー内の OID にアクセスするために、認証パスワードとプライバシー パスワードの両方を入力する必要があります。

次に、SNMP ユーザに適用する属性を確認する例を示します。

```
RP/0/RP0/CPU0:router# show snmp user

  User name: privuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

SNMP トラップ通知の設定

ここでは、SNMP トラップ通知を送信するようにルータを設定する方法について説明します。

（注） SNMPv3 の設定 （90 ページ） タスクで説明した手順をすでに完了している場合は、SNMPv3 の設定 （90 ページ） を省略できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**snmp-servergroup***name*{**v1v2v3**{**auth** | **noauth** | **priv**}}[**read***view*]**write***view*] [**notify***view*] [*access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 writer view_name2`	新規 SNMP グループ、または SNMP ユーザを SNMP ビューにマッピングするテーブルを設定します。
ステップ 3	**snmp-serveruser***groupname*{**v1v2cv3**{**auth** | **md5** | **sha**}{**clear** | **encrypted**}*auth-password*] [**priv des56** {**clear** | *access-list-name*] 例： `RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 writer view_name2`	新規 SNMP グループ、または SNMP ユーザを SNMP ビューにマッピングするテーブルを設定します。
ステップ 4	**snmp-serveruser***usernamegroupname*{**v1v2cv3**{**auth** | **md5** | **sha**}{**clear** | **encrypted**}*auth-password*] [**priv des56** {**clear** | *access-list-name*] 例： `RP/0/RP0/CPU0:routerconfig# snmp-server user noauthuser group_name v3`	新規 SNMP グループ、または SNMP ユーザを SNMP ビューにマッピングするテーブルを設定します。
ステップ 5	[**snmp-server host** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] 例： `RP/0/RP0/CPU0:router(config)# snmp-server host 12.26.25.61 traps version 3 noauth userV3noauth`	SNMP トラップ通知、使用する SNMP のバージョン、通知のセキュリティ レベル、通知の受信者（ホスト）を指定します。

	コマンドまたはアクション	目的
ステップ 6	**snmp-server traps** [*notification-type*] 例： `RP/0/RP0/CPU0:router(config)# snmp-server traps bgp`	トラップ通知の送信をイネーブルにし、送信するトラップ通知のタイプを指定します。 • トラップを *notification-type* 引数で指定しない場合は、サポートされるすべてのトラップ通知がルータ上でイネーブルになります。ルータで使用可能なトラップ通知を表示するには、**snmp-server traps ?** コマンドを入力します。
ステップ 7	**commit**	
ステップ 8	（任意） **show snmp host** 例： `RP/0/RP0/CPU0:router# show snmp host`	設定された SNMP 通知の受信者（ホスト）、ポート番号、セキュリティ モデルに関する情報を表示します。

トラップ通知の設定：例

次に、異なるタイプのトラップを送信するように SNMP エージェントを設定する例を示します。設定には、v2c ユーザ、noAuthNoPriv ユーザ、anauthNoPriv ユーザ、および AuthPriv ユーザが含まれます。

（注） デフォルトのユーザ データグラム プロトコル（UDP）ポートは 161 です。 **udp-port** キーワードおよび *port* 引数を使用して UDP ポートを指定しないと、設定された SNMP トラップ通知はポート 161 に送信されます。

```
!
  snmp-server host 10.50.32.170 version 2c userv2c udp-port 2345
  snmp-server host 10.50.32.170 version 3 auth userV3auth udp-port 2345
  snmp-server host 10.50.32.170 version 3 priv userV3priv udp-port 2345
  snmp-server host 10.50.32.170 version 3 noauth userV3noauth udp-port 2345
  snmp-server user userv2c groupv2c v2c
  snmp-server user userV3auth groupV3auth v3 auth md5 encrypted 140F0A13
  snmp-server user userV3priv groupV3priv v3 auth md5 encrypted 021E1C43 priv des56
encrypted 1110001C
  snmp-server user userV3noauth groupV3noauth v3 LROwner
  snmp-server view view_name 1.3 included
  snmp-server community public RW
  snmp-server group groupv2c v2c read view_name
  snmp-server group groupV3auth v3 auth read view_name
  snmp-server group groupV3priv v3 priv read view_name
  snmp-server group groupV3noauth v3 noauth read view_name
  !
```

次に、SNMP トラップ通知の受信者ホストの設定、つまり SNMP トラップ通知の受信者を確認する方法を示しています。出力には、次の情報が表示されます。

- 設定された通知ホストの IP アドレス
- SNMP 通知メッセージが送信される UDP ポート
- 設定されたトラップのタイプ
- 設定されたユーザのセキュリティ レベル
- 設定されたセキュリティ モデル

```
config
  show snmp host

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3auth security model: v3 auth

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3noauth security model: v3 noauth

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3priv security model: v3 priv

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userv2c security model: v2c
```

SNMP エージェントの連絡先、場所、およびシリアル番号の設定

このタスクは、SNMPエージェントのシステムの連絡先文字列、システムの場所の文字列、およびシステム シリアル番号を設定する方法について説明します。

（注） このタスクでは **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-server contact** *system-contact-string* 例： `RP/0/RP0/CPU0:router(config)# snmp-server contact Dial System Operator at beeper # 27345`	システムの連絡先文字列を設定します。
ステップ 3	（任意） **snmp-server location** *system-location* 例：	システムの場所を表す文字列を設定します。

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router(config)# snmp-server location Building 3/Room 214`	
ステップ 4	（任意） **snmp-server chassis-id** *serial-number* 例： `RP/0/RP0/CPU0:router(config)# snmp-server chassis-id 1234456`	システムのシリアル番号を設定します。
ステップ 5	**commit**	

SNMP エージェント パケットの最大サイズの定義

このタスクでは、SNMP サーバが要求を受信しているか応答を生成しているときに、許可される SNMP パケットの最大サイズを設定する例を示します。

（注） このタスクでは **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-server packetsize** *byte-count* 例： `RP/0/RP0/CPU0:router(config)# snmp-server packetsize 1024`	最大パケット サイズを設定します。
ステップ 3	**commit**	

通知操作値の変更

SNMP 通知がイネーブルになると、送信元インターフェイス、メッセージキューの長さ、または再送信間隔にデフォルト以外の値を指定することができます。

ここでは、トラップ通知用の送信元インターフェイス、各ホストのメッセージキューの長さ、および再送信間隔を指定する方法について説明します。

（注） このタスクでは **snmp-server** コマンドを実行する順序は重要ではありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	（任意） **snmp-server trap-source** *type interface-path-id* 例： `RP/0/RP0/CPU0:router(config)# snmp-server trap-source POS 0/0/1/0`	トラップ通知の送信元インターフェイスを指定します。
ステップ 3	（任意） **snmp-server queue-length** *length* 例： `RP/0/RP0/CPU0:router(config)# snmp-server queue-length 20`	各通知のメッセージ キューの長さを設定します。
ステップ 4	（任意） **snmp-server trap-timeout** *seconds* 例： `RP/0/RP0/CPU0:router(config)# snmp-server trap-timeout 20`	再送信キューにある通知を再送信する頻度を定義します。
ステップ 5	**commit**	

IP precedence および DSCP 値の設定

ここでは、SNMP トラフィックに対して IP precedence または IP DSCP を設定する方法について説明します。

始める前に

SNMP が設定されていること。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	次のいずれかのコマンドを使用します。 • **snmp-server ipv4 precedence** *value* • **snmp-server ipv4 dscp** *value* 例： `RP/0/RP0/CPU0:router(config)# snmp-server dscp 24`	SNMP トラフィックの IP precedence または IP DSCP 値を設定します。
ステップ 3	**commit**	

SNMP トラフィックの IP precedence 値の設定：例

次の例に、SNMP IP precedence 値を 7 に設定する方法を示します。

```
configure
  snmp-server ipv4 precedence 7
  exit

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
```

SNMP トラフィックの IP DSCP 値の設定：例

次の例に、SNMP トラフィックの IP DSCP 値を 45 に設定する方法を示します。

```
configure
  snmp-server ipv4 dscp 45
  exit

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
```

SNMP コンテキスト マッピングの表示

SNMPエージェントは、クライアント機能により作成されたSNMPコンテキストに基づいてクエリーを提供します。コンテキスト マッピング テーブルが存在します。コンテキスト マッピングテーブルの各エントリには、コンテキスト名、コンテキストを作成した機能の名前、および機能の特定のインスタンスの名前が含まれます。

手順

	コマンドまたはアクション	目的
ステップ 1	**show snmp context-mapping** 例： `RP/0/RP0/CPU0:router# show snmp context-mapping`	SNMP コンテキスト マッピング テーブルを表示します。

パケット損失のモニタリング

パケット損失が指定したしきい値を超えたときの SNMP トラップの生成を設定することにより、パケット損失をモニタすることが可能です。このタスクで説明する設定は、EVENT-MIB の MIB テーブルのエントリの作成をイネーブルにします。これは、その後 SNMP GET 操作を使用してパケット損失をモニタできます。

始める前に

（注） このタスクで説明する設定を使用して、EVENT-MIB MIB テーブルに作成されたエントリは、SNMP SET を使用して変更できません。

SNMP SET を使用して作成された、EVENT-MIB MIB テーブルへのエントリは、このタスクで説明する設定を使用して変更できません。

手順

	コマンドまたはアクション	目的
ステップ 1	**snmp-server mibs eventmib packet-loss** *type interface-path-id* **falling** *lower-threshold* **interval** *sampling-interval* **rising** *upper-threshold* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs eventmib packet-loss falling 1 interval 5 rising 2`	パケット損失が指定したしきい値を超えたときに、インターフェイスに対して SNMP EVENT-MIB トラップを生成します。最大 100 のインターフェイスをモニタできます。 **falling** *lower-threshold* ：低い方のしきい値を指定します。2 つの間隔間のパケット損失がこのしきい値を下回り、以前に mteTriggerRising トラップが生成されていた場合、SNMP の mteTriggerFalling トラップが生成されます。このトラップは、パケット損失が高い方のしきい値を超えて、その後、低い方のしきい値を再度下回るまで生成されません。

	コマンドまたはアクション	目的
		interval *sampling-interval* ：パケット損失の統計情報がポーリングされる頻度を指定します。これは、5 ～ 1440 分の 5 の倍数の値です。 **rising** *upper-threshold* ：高い方のしきい値を指定します。2 つの間隔間のパケット損失がこのしきい値を超えると、SNMP の mteTriggreRising トラップが生成されます。このトラップは、パケット損失が下限しきい値を下回ってから、上限しきい値を上回るまで生成されません。

維持する MIB データの設定

SNMP MIB 定義では、多くの場合、オブジェクト テーブルに任意の 32 ビットのインデックスを定義しています。MIB の実装では、多くの場合、MIB インデックスから内部データ構造へのマッピングを行います。このデータ構造は他のデータ セットのキーになります。このような MIB テーブルでは、テーブル内に含まれるデータが、モデル化されている他の要素の識別子となっている場合があります。たとえば、ENTITY-MIB においては、entPhysicalTable のエントリは 31 ビットの値である entPhysicalIndex によってインデックス化されていますが、このエントリは entPhysicalName またはテーブル内の他のオブジェクトの組み合わせによって識別することができます。

一部の MIB テーブルのサイズが原因で、32 ビット MIB インデックスから、ネットワーク管理ステーションがエントリを識別できる他のデータへのすべてのマッピングを検出するには、膨大な処理が必要になります。そのため、プロセスの再開、リスタート、スイッチオーバー、デバイスのリロードを行っても、一部の MIB インデックスが維持される必要が生じます。ENTITY-MIB の entPhysicalTable および CISCO-CLASS-BASED-QOS-MIB は、このような MIB の例であり、インデックス値を維持する必要が生じる場合が多くあります。

また、CISCO-CLASS-BASED-QOS-MIB 統計情報のクエリ実行時のクエリの応答時間や CPU 使用率の問題により、サービス ポリシーの統計情報はキャッシュしておくことが望ましいと言えます。

手順

	コマンドまたはアクション	目的
ステップ 1	（任意） **snmp-server entityindex persist** **例：** `RP/0/RP0/CPU0:router(config)# snmp-server entityindex persist`	ENTITY-MIB データの固定ストレージをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 2	（任意） **snmp-server mibs cbqosmib persist** 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib persist`	CISCO-CLASS-BASED-QOS-MIB データの固定ストレージをイネーブルにします。
ステップ 3	（任意） **snmp-server cbqosmib cache refresh time** *time* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib cache refresh time 45`	QoS MIB のキャッシュをイネーブルにして、キャッシュのリフレッシュ時間を設定します。
ステップ 4	（任意） **snmp-server cbqosmib cache service-policy count** *count* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib cache service-policy count 50`	QoS MIB のキャッシュをイネーブルにして、キャッシュするサービスポリシーの数に制限を設けます。
ステップ 5	**snmp-server ifindex persist** 例： `RP/0/RP0/CPU0:router(config)# snmp-server ifindex persist`	すべての簡易ネットワーク管理プロトコル（SNMP）インターフェイスで、ifIndex パーシステンスをグローバルにイネーブルにします。

インターフェイスのサブセットに対する linkUp および linkDown トラップの設定

トラップを設定するインターフェイスを表すための正規表現を指定することで、同時に多数のインターフェイスに対して linkUp および linkDown トラップをイネーブルまたはディセーブルにすることができます。

始める前に

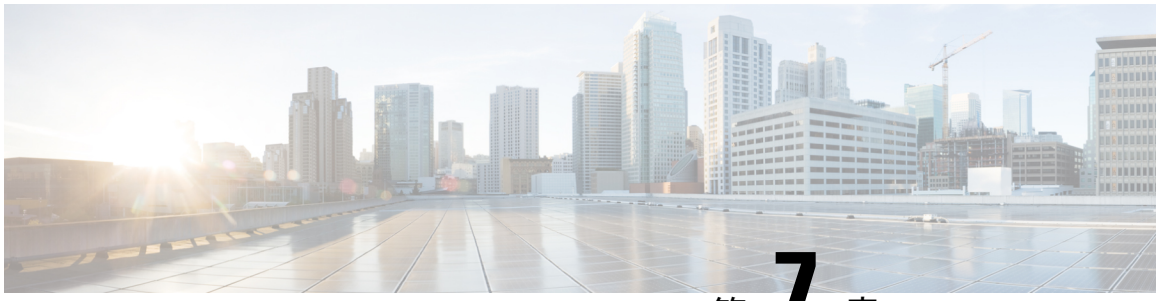
SNMP が設定されていること。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**snmp-server interface subset** *subset-number* **regular-expression** *expression* 例： `RP/0/RP0/CPU0:router(config)# snmp-server interface subset 10 regular-expression "^Gig[a-zA-Z]+[0-9/]+\." RP/0/RP0/CPU0:router(config-snmp-if-subset)#`	正規表現で識別されたインターフェイスに対し、snmp-server インターフェイスモードを開始します。 subset-number 引数は、インターフェイスのセットを識別し、インターフェイスが複数のサブセットに含まれている場合は、そのサブセットのプライオリティも割り当てます。数値が小さいほどプライオリティが高く、そのコンフィギュレーションは数値が大きいインターフェイスサブセットよりも優先されます。 *expression* 引数は二重引用符で囲んで入力する必要があります。 正規表現の詳細については、 の「*Understanding Regular Expressions, Special Characters, and Patterns*」モジュールを参照してください。
ステップ 3	**notification linkupdown disable** 例： `RP/0/RP0/CPU0:router(config-snmp-if-subset)# notification linkupdown disable`	設定しているすべてのインターフェイスに対して linkUp および linkDown トラップをディセーブルにします。ディセーブルにしたインターフェイスをイネーブルにするには、このコマンドの **no** 形式を使用します。
ステップ 4	**commit**	
ステップ 5	（任意） **show snmp interface notification subset** *subset-number* 例： `RP/0/RP0/CPU0:router# show snmp interface notification subset 10`	サブセットのプライオリティで識別されたすべてのインターフェイスについて、linkUp および linkDown 通知のステータスを表示します。
ステップ 6	（任意） **show snmp interface notification regular-expression** *expression* 例： `RP/0/RP0/CPU0:router# show snmp interface notification regular-expression "^Gig[a-zA-Z]+[0-9/]+\."`	正規表現で識別されたすべてのインターフェイスについて、linkUp および linkDown 通知のステータスを表示します。

	コマンドまたはアクション	目的
ステップ **7**	（任意） **show snmp interface notification** *type interface-path-id* 例： `RP/0/RP0/CPU0:router# show snmp interface notification tengige 0/11/0/0.10`	指定されたインターフェイスについて、linkUp および linkDown 通知のステータスを表示します。



第 7 章

定期的な MIB データの収集および転送の設定

このマニュアルでは、選択された MIB データをルータから指定されたネットワーク管理システム（NMS）に定期的に転送する方法について説明します。定期的な MIB データ収集と転送機能は、バルク統計情報とも呼ばれます。

- 定期的な MIB データの収集および転送の前提条件 （109 ページ）
- 定期的な MIB データの収集および転送に関する情報 （109 ページ）
- 定期的な MIB データの収集および転送の設定方法 （111 ページ）
- 定期的な MIB データの収集および転送：例 （117 ページ）

定期的な MIB データの収集および転送の前提条件

定期的な MIB データ収集と転送を使用するには、管理情報の簡易ネットワーク管理プロトコル（SNMP）モデルに精通している必要があります。また、ネットワーク デバイスでモニタする MIB 情報や、モニタ対象である MIB オブジェクトの OID またはオブジェクト名を知っている必要があります。

定期的な MIB データの収集および転送に関する情報

SNMP のオブジェクトとインスタンス

SNMP 管理情報のタイプ（またはクラス）をオブジェクトと呼びます。管理情報のタイプの特定のインスタンスをオブジェクト インスタンス（または SNMP 変数）と呼びます。バルク統計情報収集を設定するには、バルク統計情報オブジェクト リストを使用してモニタするオブジェクト タイプと、バルク統計情報スキーマを使用して収集するオブジェクトの特定のインスタンスを指定する必要があります。

オブジェクト識別子（OID）と呼ばれる一連の番号を使用すると、MIB、MIB テーブル、MIB オブジェクト、オブジェクトのインデックスをすべて指定できます。OID は、バルク統計情報

オブジェクトリスト（一般的なオブジェクト用）とバルク統計情報スキーマ（特定のオブジェクト インスタンス用）の両方のバルク統計情報収集の設定に使用されます。

バルク統計情報オブジェクト リスト

ポーリング対象の MIB オブジェクトをグループ化するには、1 つまたは複数のオブジェクト リストを作成する必要があります。バルク統計情報オブジェクトリストは、同じMIBインデックスを共有する、ユーザ指定の MIB オブジェクトのセットです。オブジェクト リストは、指定した名前によって識別されます。名前付きのバルク統計情報オブジェクトリストを使用すると、異なるバルク統計情報スキーマで同じ設定を再利用できます。

オブジェクト リストのオブジェクトはすべて、同じ MIB インデックスを共有する必要があります。ただし、オブジェクトが同じ MIB 内に存在したり、同じ MIB テーブルに属する必要はありません。たとえば、ifInOctets と CISCO-IF-EXTENSION-MIB オブジェクトを同じスキーマでグループ化することが可能です。これは、両方のオブジェクトに対して含まれているテーブルが ifIndex によって指標付けされるためです。

バルク統計情報スキーマ

定期的な MIB データの収集および転送のメカニズムに対するデータの選択には、次の情報を含むスキーマの定義が必要です。

- オブジェクト リストの名前。
- 指定されたオブジェクトリスト内で取得する必要があるオブジェクトのインスタンス（ワイルド カードを使用して定義された特定のインスタンスまたは一連のインスタンス）。
- 指定したインスタンスに対して必要なサンプリングの頻度（ポーリング間隔）。デフォルトのポーリング間隔は 5 分です。

バルク統計情報スキーマも、指定した名前によって識別されます。この名前は、転送オプションを設定する際に使用されます。

バルク統計情報転送オプション

収集するデータを設定した後、収集した全データを使用して単一の仮想ファイル（VFile またはバルク統計情報ファイル）が作成されます。このファイルは、FTP または TFTP を使用してネットワーク管理ステーションに転送できます。このファイルの転送頻度を指定できます。デフォルトの転送間隔は、30 分に 1 回です。何らかの理由でプライマリ ネットワーク管理ステーションに転送できない場合に使用されるセカンダリ宛先を設定することもできます。

転送間隔の値は、ローカルのバルク統計情報ファイルの収集期間（収集間隔）でもあります。収集期間が終了すると、そのバルク統計情報ファイルは凍結し、データを格納するためにローカルのバルク統計情報ファイルが新たに作成されます。その後、凍結したバルク統計情報ファイルは指定した宛先に転送されます。

デフォルトでは、ローカルのバルク統計情報ファイルは、ネットワーク管理ステーションに正常に転送された後に削除されます。

定期的な MIB データの収集および転送の利点

定期的な MIB データの収集および転送（バルク統計情報機能）では、バルク ファイル MIB（CISCO-BULK-FILE-MIB.my）と同じ機能の多くを使用できますが、重要な利点がいくつかあります。主な利点は、この機能が CLI を使用して設定でき、外部のモニタリング アプリケーションが不要なことです。

定期的な MIB データの収集および転送では、バルク統計情報ファイルの保存は主に（揮発性または永続的な）ローカル ストレージが十分にある中規模からハイエンドのプラットフォームをターゲットとしています。バルク統計情報ファイルをローカルに保存すると、一時的なネットワーク停止時にデータ損失を最小限に抑えられます。

この機能にはバルク ファイル MIB よりも強力なデータ選択機能があるため、異なるテーブルの MIB オブジェクトをデータ グループ（オブジェクト リスト）にグループ化することも可能です。また、この機能はより柔軟性のあるインスタンス選択メカニズムを備えています。このメカニズムでは、アプリケーションは MIB テーブル全体を取得するように制限されていません。

定期的な MIB データの収集および転送の設定方法

バルク統計情報オブジェクト リストの設定

定期的な MIB データの収集および転送のメカニズムを設定する場合の最初の手順は、1 つまたは複数のオブジェクト リストを設定することです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**snmp-server mib bulkstat object-list** *list-name* **例**： `snmp-server mib bulkstat object-list ifMib`	SNMP バルク統計情報オブジェクト リストを定義し、バルク統計情報オブジェクト リスト コンフィギュレーション モードを開始します。

ステップ 3	**add** {**oid** | *object-name*}
	例：
	`RP/0/RP0/CPU0:router(config-bulk-objects)# add 1.3.6.1.2.1.2.2.1.11`
	`RP/0/RP0/CPU0:router(config-bulk-objects)#`
		MIB オブジェクトをバルク統計情報オブジェクト リストに追加します。モニタ対象の全オブジェクトがこのリストに追加されるまで、必要に応じて繰り返します。

	コマンドまたはアクション	目的
	add ifAdminStatus RP/0/RP0/CPU0:router(config-bulk-objects)# add ifDescr	（注） バルク統計情報オブジェクト リスト内のオブジェクトはすべて、同じ MIB インデックスによって指標付けされる必要があります。ただし、オブジェクト リスト内のオブジェクトが同じ MIB または MIB テーブルに属する必要はありません。 OID ではなくオブジェクト名を指定すると（add コマンドを使用）、**show snmp mib object** コマンドの出力で示されるマッピングのあるオブジェクト名だけを使用できます。
ステップ 4	**commit**	

バルク統計情報スキーマの設定

定期的な MIB データの収集および転送を設定する場合の 2 つめの手順は、1 つまたは複数のスキーマを設定することです。

始める前に

スキーマで使用されるバルク統計情報オブジェクト リストを定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**snmp-server mib bulkstat schema** *schema-name* 例： RP/0/RP0/CPU0:router(config)# snmp-server mib bulkstat schema intE0 RP/0/RP0/CPU0:router(config-bulk-sc)#	バルク統計情報スキーマを指定し、バルク統計情報スキーマ モードを開始します。
ステップ 3	**object-list** *list-name* 例：	このスキーマに含めるバルク統計情報オブジェクトリストを指定します。スキーマごとにオブジェクト リストを 1 つだ

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router(config-bulk-sc)# object-list ifMib`	け指定してください。複数の object-list コマンドが実行されると、より新しいコマンドによって先行のコマンドが上書きされます。

ステップ 4	次のいずれかを実行します。
	• **instance exact** { **interface** *interface-id* [**sub-if**] | **oid** *oid*}
	• **instance wild** { **interface** *interface-id* [**sub-if**] | **oid** *oid*}
	• **instance range start** *oid* **end** *oid*
	• **instance repetition** *oid* **max** *repeat-number*
	例：
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance wild oid 1`
	または
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance exact interface TenGigE 0/1.25`
	または
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance range start 1 end 2`
	または
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance repetition 1 max 4`
		このスキーマにおけるオブジェクトのインスタンス情報を指定します。
		• **instance exact** コマンドは、指定されたインスタンスが完全な OID であることを示しています（オブジェクト リストに追加されている場合）。
		• **instance wild** コマンドは、指定した OID のすべてのサブインデックスがこのスキーマに属することを示しています。wild キーワードを使用すると、部分的に「ワイルドカードを使用した」インスタンスを指定できます。
		• **instance range** コマンドは、データを収集するインスタンスの範囲を示します。
		• **instance repetition** コマンドは、MIB オブジェクトの特定の数のインスタンスに対して繰り返すデータ収集を示します。
		（注） 1 つのスキーマに設定できる **instance** コマンドは 1 つだけです。複数の **instance** コマンドが実行されると、新しいコマンドによって先行のコマンドが上書きされます。

ステップ 5	**poll-interval** *minutes* **例**： `RP/0/RP0/CPU0:router(config-bulk-sc)# poll-interval 10`	このスキーマで指定されたオブジェクト インスタンスからデータを収集する頻度を分単位で設定します。デフォルトは、5 分に 1 回です。有効範囲は 1 ～ 20000 です。
ステップ 6	**commit**	

バルク統計情報転送オプションの設定

定期的な MIB データの収集および転送を設定する最後の手順は、転送オプションを設定することです。収集された MIB データは、VFile（仮想ファイル。このマニュアル内ではバルク統計情報ファイルとも呼ばれている）と呼ばれるローカルファイルのようなエンティティに格納されます。このファイルは、ユーザが指定した間隔でリモートのネットワーク管理ステーションに転送できます。

始める前に

バルク統計情報オブジェクト リストとバルク統計情報スキーマは、バルク統計情報転送オプションを設定する前に定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**snmp-server mib bulkstat transfer-id** *transfer-id* 例： `RP/0/RP0/CPU0:router(config)# snmp-server mib bulkstat transfer bulkstat1`	転送設定を名前（*transfer-id* 引数）で識別し、バルク統計情報転送コンフィギュレーションモードを開始します。
ステップ 3	**buffer-size** *bytes* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# buffersize 3072`	（任意）バルク統計情報データ ファイルの最大サイズをバイト単位で指定します。有効範囲は 1024 ～ 2147483647 バイトです。デフォルトのバッファ サイズは 2048 バイトです。 （注） 転送間隔時間が切れる前に、バルク統計情報ファイルの最大バッファ サイズに到達した場合、追加で受信したすべてのデータが削除されます。この動作を修正するために、ポーリング頻度を減らしたり、バルク統計情報バッファのサイズを増やせます。
ステップ 4	例：	（任意）バルク統計情報データ ファイル（VFile）の形式を指定します。デフォルトは schemaASCII です。

	コマンドまたはアクション	目的
		（注） 転送を実行できるのは、schemaASCII（cdcSchemaASCII）形式を使用した場合に限ります。SchemaASCII は、データ値を解析するためのパーサーフレンドリなヒントを含むヒト可読形式です。

ステップ 5	**schema** *schema-name*
	例：
	`RP/0/RP0/CPU0:router(config-bulk-tr)# schema TenGigE 0/11/0/0`
	`RP/0/RP0/CPU0:router(config-bulk-tr)# schema TenGigE/0-CAR`
	`RP/0/RP0/CPU0:router(config-bulk-tr)# schema TenGigE 0/11/0/0`
		転送するバルク統計情報スキーマを指定します。必要に応じて、このコマンドを繰り返します。複数のスキーマを単一の転送設定に関連付けることができます。収集された全データが単一のバルク データ ファイル（VFile）に保存されます。

ステップ 6	**transfer-interval** *minutes* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# transfer-interval 20`	（任意）バルク統計情報ファイルの転送頻度を分単位で指定します。デフォルト値は、30 分に 1 回です。転送間隔は、収集間隔と同じです。
ステップ 7	**url primary** *url* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# url primary ftp://user:password@host/folder/bulkstat1`	バルク統計情報データ ファイルを転送するネットワーク管理システム（ホスト）と転送に使用するプロトコルを指定します。この宛先は、ユニフォーム リソース ロケータ（URL）として指定されます。FTP または TFTP は、バルク統計情報ファイルの転送に使用できます。
ステップ 8	**url secondary** *url* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# url secondary tftp://10.1.0.1/tftpboot/user/bulkstat1`	（任意）プライマリ ロケーション ファイルへの転送イベントで使用する、バックアップの転送先とプロトコルを指定します。FTP または TFTP は、バルク統計情報ファイルの転送に使用できます。
ステップ 9	**retry** *number* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# retry 1`	（任意）送信の再試行回数を指定します。デフォルト値は 0（つまり、再試行しない）です。バルク統計情報ファイルを送信しようとして失敗した場合に、このコマンドを使用してファイル

	コマンドまたはアクション	目的
		の再送信を試みるように設定できます。 1 回の再試行に含まれるのは、まず 1 番目の宛先に試行され、転送に失敗すると、次に 2 番目の場所に試行される動作です。たとえば、再試行値が 1 の場合、まずプライマリ URL に試行された後でセカンダリ URL に、そして再びプライマリ URL への後で再びセカンダリ URL に試行されます。有効範囲は 0 ～ 100 です。 すべての再試行に失敗すると、次の通常の転送は、設定された転送間隔の時間が経過した後に実行されます。
ステップ 10	**retain** *minutes* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# retain 60`	（任意）収集間隔と転送試行の設定が完了したら、バルク統計情報ファイルをシステムメモリに保存する期間を分単位で指定します。デフォルト値は 0 です。ゼロ（0）は、転送が試行された直後にファイルが削除されることを示します。有効範囲は 0 ～ 20000 です。 （注） retry コマンドを使用する場合、保持間隔を 0 よりも大きく設定する必要があります。再試行の間隔は、保持間隔を再試行回数で割ったものです。たとえば、**retain 10** および **retry 2** が設定されている場合は、2 回の再試行が 5 分ごとに 1 回試行されます。したがって、retain 0 に設定した場合、再試行は行われません。
ステップ 11	**enable** 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# enable`	この設定でバルク統計情報のデータ収集と転送のプロセスを開始します。 • このアクションを正常に実行させるには、オブジェクト数が 0 以外のスキーマを最低 1 つは設定する必要があります。

	コマンドまたはアクション	目的
		• 定期的なデータ収集とファイル転送が開始されるのは、このコマンドが設定されている場合だけです。逆に、**no enable** コマンドが設定されていると、収集プロセスが停止します。後続の **enable** では、動作が再び開始されます。 • **enable** コマンドを使用して収集プロセスが開始されるたびに、新しいバルク統計情報ファイルにデータが収集されます。**no enable** コマンドを使用すると、収集したデータの転送プロセスがただちに開始されます（つまり、既存のバルク統計情報ファイルが指定した管理ステーションに転送されます）。
ステップ 12	**commit** *minutes* 例： `RP/0/RP0/CPU0:router(config-bulk-tr)# retain 60`	転送間隔時間が切れる前に、バルク統計情報ファイルの最大バッファ サイズに到達した場合でも、転送動作は開始されますが、ファイルがいっぱいになった後で受信したバルク統計情報のデータで転送される前のものは削除されます。この動作を修正するために、ポーリング頻度を減らしたり、バルク統計情報バッファのサイズを増やせます。 **retain 0** に設定した場合、再試行は行われません。これは、再試行の間隔が retain の値を retry の値で割ったものであるためです。たとえば、**retain 10** および **retry 2** が設定されている場合は、再試行が 5 分ごとに 1 回行われます。したがって、retry コマンドを設定した場合、retain コマンドにも適切な値を設定する必要があります。

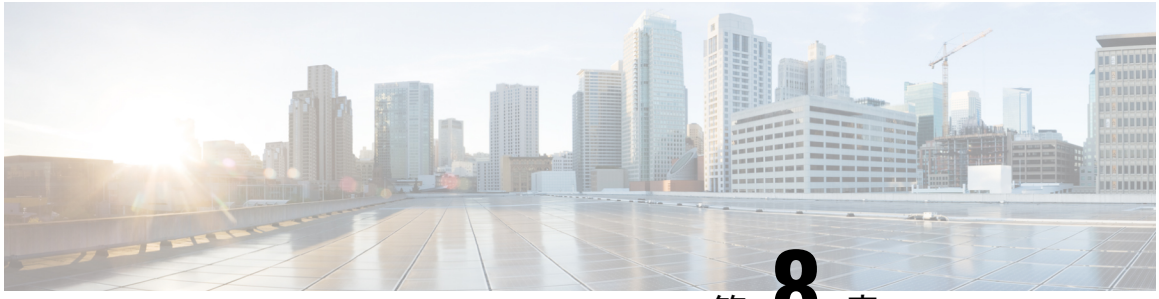
定期的な MIB データの収集および転送：例

次に、定期的な MIB データ収集および転送を設定する例を示します。

```
snmp-server mib bulkstat object-list cempo
add cempMemPoolName
add cempMemPoolType
!
snmp-server mib bulkstat schema cempWild
object-list cempo
instance wild oid 8695772
poll-interval 1
!
snmp-server mib bulkstat schema cempRepeat
object-list cempo
instance repetition 8695772.1 max 4294967295
poll-interval 1
!
snmp-server mib bulkstat transfer-id cempt1
enable
url primary tftp://223.255.254.254/auto/tftp-sjc-users3/username/dumpdcm
schema cempWild
schema cempRepeat
transfer-interval 2
!
```

次に、バルク統計情報ファイルの内容の例を示します。

```
Schema-def cempt1.cempWild "%u, %s, %s, %d" Epochtime instanceoid
            1.3.6.1.4.1.9.9.221.1.1.1.1.3 1.3.6.1.4.1.9.9.221.1.1.1.1.2
cempt1.cempWild: 1339491515, 8695772.1, processor, 2
cempt1.cempWild: 1339491515, 8695772.2, reserved, 11
cempt1.cempWild: 1339491515, 8695772.3, image, 12
cempt1.cempWild: 1339491575, 8695772.1, processor, 2
cempt1.cempWild: 1339491575, 8695772.2, reserved, 11
cempt1.cempWild: 1339491575, 8695772.3, image, 12
Schema-def cempt1.cempRepeat "%u, %s, %s, %d" Epochtime instanceoid
            1.3.6.1.4.1.9.9.221.1.1.1.1.3 1.3.6.1.4.1.9.9.221.1.1.1.1.2
cempt1.cempRepeat: 1339491515, 8695772.1, processor, 2
cempt1.cempRepeat: 1339491515, 8695772.2, reserved, 11
cempt1.cempRepeat: 1339491515, 8695772.3, image, 12
cempt1.cempRepeat: 1339491515, 26932192.1, processor, 2
cempt1.cempRepeat: 1339491515, 26932192.2, reserved, 11
cempt1.cempRepeat: 1339491515, 26932192.3, image, 12
cempt1.cempRepeat: 1339491515, 35271015.1, processor, 2
cempt1.cempRepeat: 1339491515, 35271015.2, reserved, 11
cempt1.cempRepeat: 1339491515, 35271015.3, image, 12
cempt1.cempRepeat: 1339491515, 36631989.1, processor, 2
cempt1.cempRepeat: 1339491515, 36631989.2, reserved, 11
cempt1.cempRepeat: 1339491515, 36631989.3, image, 12
cempt1.cempRepeat: 1339491515, 52690955.1, processor, 2
cempt1.cempRepeat: 1339491515, 52690955.2, reserved, 11
cempt1.cempRepeat: 1339491515, 52690955.3, image, 12
```



第 8 章

Cisco Discovery Protocol の設定

Cisco Discovery Protocol（CDP）は、ルータ、ブリッジ、アクセス サーバ、コミュニケーション サーバ、スイッチを含め、シスコ製のあらゆる機器で動作する、メディアにもプロトコルにも依存しないプロトコルです。CDP を使用して、デバイスに直接接続しているすべてのシスコの装置の情報を表示することができます。

- CDP の実装の前提条件 （119 ページ）
- CDP の実装について （119 ページ）
- CDP の実装方法：Cisco IOS XR ソフトウェア （121 ページ）
- CDP の実装の設定例 （124 ページ）

CDP の実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

CDP の実装について

CDP の主な用途は、隣接するデバイスのプロトコル アドレスを取得し、そのデバイスのプラットフォームを検出することです。また、CDP を使用して、ルータが使用するインターフェイスの情報を表示できます。CDP はメディアとプロトコルに依存せず、ルータ、ブリッジ、アクセス サーバ、スイッチなど、シスコ製のすべての機器で実行できます。

SNMP と CDP MIB を併用すると、ネットワーク管理アプリケーションは、隣接するデバイスのデバイス タイプと SNMP エージェント アドレスを認識し、それらのデバイスに SNMP クエリーを送信できます。CDP は CISCO-CDP-MIB を使用します。

CDP は、LAN、フレーム リレー、ATM 物理メディアなど、サブネットワーク アクセス プロトコル（SNAP）をサポートするすべてのメディアで実行されます。CDP の動作はデータリンク層上に限定されます。そのため、異なるネットワーク層プロトコルをサポートする 2 つのシステムが、相互について認識できます。

CDP 用に設定された各デバイスからマルチキャスト アドレスに対してメッセージが定期的に送信されます。このメッセージはアドバタイズメントと呼ばれます。 各デバイスは、SNMP メッセージを受信できるアドレスを少なくとも 1 つアドバタイズします。アドバタイズには、存続可能時間（保持時間）や情報も含まれています。これは、受信側のデバイスが CDP 情報を破棄せずに保持する時間の長さを示します。各デバイスは、他のデバイスから送信される定期的な CDP メッセージを待ち受けます。これは、隣接するデバイスについて認識し、メディアに対するインターフェイスがアップまたはダウンした場合を判断するためです。

CDP Version-2（CDPv2）は、このプロトコルの最新リリースで、より高度なデバイス追跡機能を備えています。たとえば、より高速なエラー追跡が可能なレポート メカニズムなどが含まれるため、コストがかかるダウンタイムを減らすことができます。レポートされるエラー メッセージは、コンソールまたはロギング サーバに送信でき、また接続ポートの一致していないネイティブ VLAN ID（IEEE 802.1Q）インスタンス、および接続デバイス間の一致していないポート デュプレックス ステートをカバーできます。

CDPv2 **show** コマンドを実行すると、隣接するデバイスの VLAN トランキング プロトコル（VTP）管理ドメインとデュプレックス モード、CDP 関連のカウンタ、および接続ポートの VLAN ID に関する詳細な情報が出力されます。

Type-Length-Value（TLV）フィールドは、CDP アドバタイズメントに埋め込まれる情報ブロックです。次の表に、CDP アドバタイズメントの TLV 定義の概要を示します。

表 *4 : CDPv2* の *Type-Length-Value* 定義

TLV	定義
デバイス ID TLV	文字列形式のデバイス名を識別します。
アドレス TLV	受信デバイスと送信デバイス両方のネットワーク アドレス リストを含めます。
ポート ID TLV	CDP パッケージが送信されるポートを指定します。
機能 TLV	スイッチなど、デバイス タイプの形式でデバイスの機能を説明します。
バージョン TLV	デバイスが実行しているソフトウェア リリース バージョンに関する情報を含めます。
プラットフォーム TLV	Cisco 4500 など、デバイスのハードウェア プラットフォーム名を記述します。
VTP 管理ドメイン TLV	システムの設定済み VTP 管理ドメイン名の文字列をアドバタイズします。隣接するネットワーク ノードの VTP ドメイン コンフィギュレーションを確認するために、ネットワーク オペレータが使用します。

TLV	定義
ネイティブ VLAN TLV	インターフェイス上の非タグ付きパケットに対して想定される VLAN をインターフェイス単位で示します。CDP はインターフェイスのネイティブ VLAN を認識します。この機能を実装するのは、IEEE 802.1Q プロトコルをサポートするインターフェイスの場合だけです。
全二重/半二重 TLV	CDPブロードキャストインターフェイスのステータス（デュプレックス設定）を示します。ネットワーク オペレータが、隣接するネットワーク要素間の接続の問題を診断するときに使用します。

CDP の実装方法：Cisco IOS XR ソフトウェア

CDP のイネーブル化

CDP をイネーブルにするには、まずルータで CDP をグローバルにイネーブルにしてから、インターフェイス単位で CDP をイネーブルにする必要があります。ここでは、ルータ上で CDP をグローバルにイネーブルにし、次にインターフェイスで CDP をイネーブルにする方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**cdp** 例： `RP/0/RP0/CPU0:router# cdp`	CDP をグローバルにイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RP0/CPU0:router# int TenGigE 0/11/0/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**cdp** 例： `RP/0/RP0/CPU0:router(config-if)# int TenGigE 0/11/0/0`	特定のインターフェイス上で CDP をイネーブルにします。
ステップ 5	**commit**	

CDP デフォルト設定の変更

ここでは、デフォルトのバージョン、保持時間の設定、およびタイマーの設定を変更する方法について説明します。

（注） コマンドは任意の順序で入力できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**cdp advertise v1** 例： `RP/0/RP0/CPU0:router# cdp advertise v1`	近接装置との通信にバージョン 1（CDPv1）だけを使用するように CDP を設定します。 • デフォルトでは、CDP をイネーブルにすると、ルータから CDPv2 パケットが送信されます。相手先のデバイスで CDPv2 パケットが処理されない場合は、CDPv1 パケットも送受信されます。 • この例では、ルータが CDPv1 パケットだけを送受信するよう設定されています。
ステップ 3	**cdp holdtime** *seconds* 例： `RP/0/RP0/CPU0:router# cdp holdtime 30`	ネットワーク デバイスがルータから送信された CDP パケットを受信した後、破棄するまで保持する時間の長さを指定します。 • デフォルトでは、CDP がイネーブルの場合、受信ネットワーキング デバイスは、CDP パケットを廃棄するまでに 180 秒間保持します。 （注） CDP 保持時間は、CDP の送信間隔（**cdp timer** コマンドを使用して設定します）よりも長い秒数に設定する必要があります。 • この例では、*seconds* 引数の保持時間の値が 30 に設定されます。

	コマンドまたはアクション	目的
ステップ 4	**cdp timer** *seconds* 例： `RP/0/RP0/CPU0:router# cdp timer 20`	CDP アップデート パケットが送信される頻度を指定します。 • デフォルトでは、CDP がイネーブルになっている場合、CDP 更新パケットが 60 秒ごとに 1 回の頻度で送信されます。 （注） タイマーの設定時間が短いほど、CDP 更新の送信頻度が高くなります。 • この例では、CDP 更新パケットが 20 秒ごとに 1 回の頻度で送信されるように設定されます。
ステップ 5	**commit**	
ステップ 6	（任意） **show cdp** 例： `RP/0/RP0/CPU0:router# show cdp`	グローバルな CDP 情報を表示します。 出力には、ルータで実行中の CDP バージョン、保持時間の設定、およびタイマー設定が表示されます。

CDP のモニタリング

このタスクでは、CDP をモニタする例を示します。

（注） コマンドは任意の順序で入力できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**show cdp entry** {***** | *entry-name*} [**protocol** | **version**] 例： `RP/0/RP0/CPU0:router# show cdp entry *`	CDP を使用して検出された特定の近接装置またはすべての近接装置に関する情報を表示します。
ステップ 2	**show cdp interface** [*type interface-path-id* | **location** *node-id*] 例：	CDP をイネーブルにしたインターフェイスに関する情報を表示します。

	コマンドまたはアクション	目的
	`RP/0/RP0/CPU0:router# show cdp interface pos 0/0/0/1`	
ステップ 3	**show cdp neighbors** [*type interface-path-id* | **location** *node-id*] [**detail**] 例： `RP/0/RP0/CPU0:router# show cdp neighbors`	CDP を使用して検出された隣接するデバイスに関する詳細情報を表示します。
ステップ 4	**show cdp traffic** [**location** *node-id*] 例： `RP/0/RP0/CPU0:router# show cdp traffic`	CDP を使用してデバイス間で収集されたトラフィックに関する情報を表示します。

CDP の実装の設定例

CDP のイネーブル化：例

次の例に、CDP をグローバルに設定し、イーサネット インターフェイス TenGigE 0/11/0/0 上で CDP をイネーブルにする方法を示します。

```
cdp
interface 0/11/0/0
cdp
```

グローバル CDP 設定の変更：例

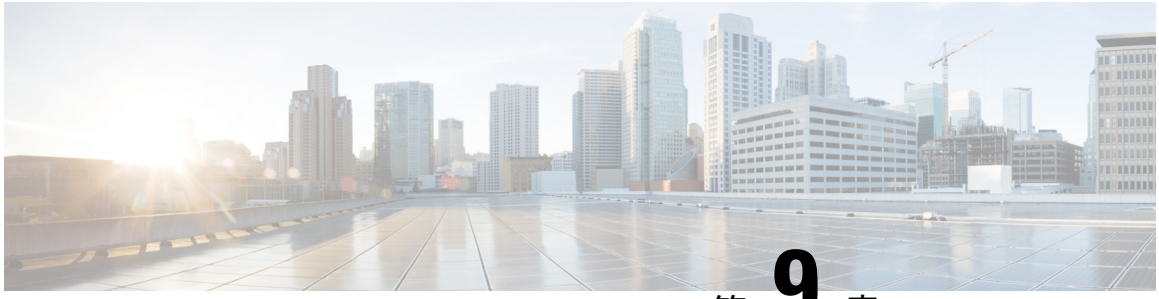
次に、グローバル CDP 設定を変更する例を示します。この例では、タイマー設定は 20 秒、ホールド時間は 30 秒、および隣接デバイスとの通信に使用する CDP のバージョンは CDPv1 に設定されています。

```
cdp timer 20
cdp holdtime 30
cdp advertise v1
```

次の例に、 **show cdp** コマンドを使用して CDP グローバル設定を確認する方法を示します。

```
RP/0/RP0/CPU0:router# show cdp

  Global CDP information:
  Sending CDP packets every 20 seconds
  Sending a holdtime value of 30 seconds
```

```
Sending CDPv2 advertisements is not enabled
```



第 9 章

スマート ライセンスを使用したライセンスの設定

- スマート ライセンスとは? （127 ページ）
- スマート ライセンスの仕組み （128 ページ）
- スマート ライセンスの導入オプション （129 ページ）
- Call Home について （131 ページ）
- サポート対象の柔軟な消費モデル ライセンス （131 ページ）
- スマート ライセンスを使用したライセンスの設定 （132 ページ）
- スマート ライセンスのワークフロー （137 ページ）
- ライセンス、製品インスタンス、および登録トークン （137 ページ）

スマート ライセンスとは?

スマート ライセンスは、時間のかかる手動のライセンス タスクを自動化できるクラウドベースのソフトウェア ライセンス管理ソリューションです。このソリューションを使用すると、ライセンスのステータスとソフトウェアの使用傾向を簡単に追跡できます。

シスコ スマート ライセンスを使うと次の 3 つのコア機能が簡素化されます。

- **購入**：ネットワークにインストールされているソフトウェアを、製品アクティベーション キー（PAK）を指定せずに自動的に登録できます。
- **管理**：ライセンス エンタイトルメントの有効化を自動的に追跡できます。また、すべてのノードにライセンス ファイルをインストールする必要はありません。組織構造に合わせたライセンス プール（ライセンスの論理的なグループ）を作成できます。Smart Licensing には、すべての Cisco ソフトウェア ライセンスを 1 つの一元化された Web サイトで管理できる集中型ポータルである Cisco Smart Software Manager が用意されています。*Cisco Smart Software Manager* に詳細が記載されています。
- **レポート**：スマート ライセンスでは、ポータルを使用することで、購入したライセンスとネットワークに展開された製品を統合して表示できます。このデータを使用すると、購入の意思決定を実際の使用状況に基づいてより適切に行うことができます。

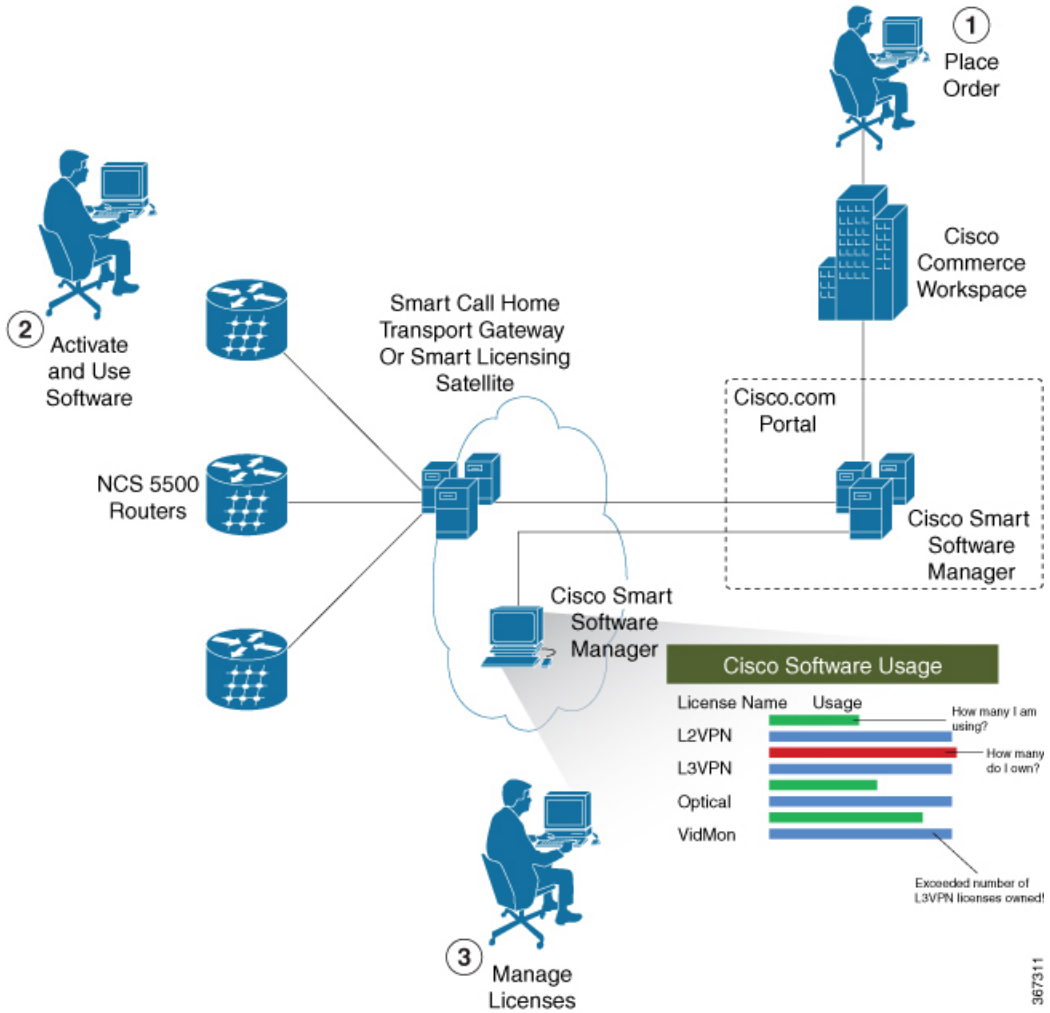
（注）

- デフォルトでは、スマート ライセンスが有効になっています。
- 非消費モデルのスマート ライセンスのみがサポートされています。

スマート ライセンスの仕組み

次の図で示しているように、スマートライセンスの動作モデルを示す3つのステップがあります。

図 *4:* スマート ライセンス：例



- **スマート ライセンスの設定**：スマート ライセンスを注文します。ライセンスはCisco.comポータルで管理します。Smart Software Manager ポータルでのスマート ライセンスの使用およびアクセスについて規定した利用規約に同意します。

- **スマート ライセンスの有効化と使用**：手順に従ってスマート ライセンスを有効にします。「スマート ライセンスのワークフロー」に手順が記載されています。

  スマート ライセンスを有効にしたら、次のいずれかのオプションを使用して通信できます。

  - **Smart Call Home**：Smart Call Home 機能は、スマート ライセンスが有効になった後に自動的に設定されます。Smart Call Home は、シスコのライセンス サービスとの通信用メディアとしてスマート ライセンスで使用されます。Call Home 機能を使用すると、シスコ製品が定期的に call-home を実行し、ソフトウェアの使用状況情報を監査および調整できるようになります。この情報により、シスコは、インストールベースを効率的に追跡し、それらを継続的に維持および稼働させることができます。また、お客様の手を煩わせることなく、サービスの提供および契約更新に関するサポートをより効果的に行えるようになります。Smart Call Home 機能の詳細については、『Smart Call Home Deployment Guide』を参照してください。

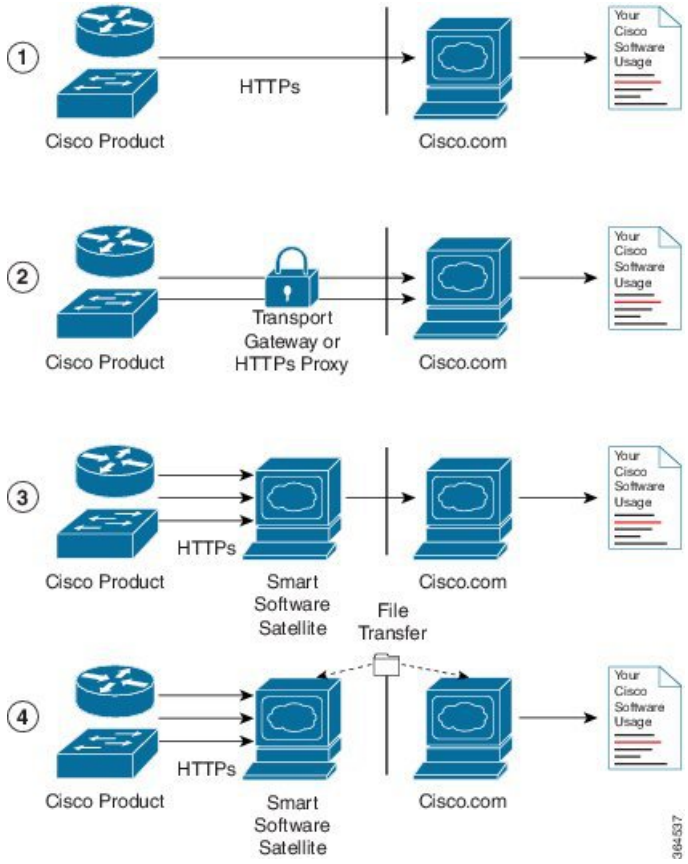
  - **スマート ライセンス サテライト**：スマート ライセンス サテライト オプションは、スマート ライセンスの使用状況を統合および管理するために使用できるオンプレミス コレクタを提供し、Cisco.com でのシスコ ライセンス サービスとの通信を促進します。

- **ライセンスの管理およびレポート**：Smart Software Manager ポータルで、ソフトウェア全体の使用状況に関するレポートを管理および表示できます。コンプライアンス レポート（138 ページ） では、スマート ライセンス レポートのタイプについて説明します。

スマート ライセンスの導入オプション

次の図は、スマート ライセンスの導入に使用できるさまざまなオプションを示しています。

図 *5:* スマート ライセンスの導入オプション



1. **ダイレクト クラウド アクセス**：ダイレクト クラウド アクセスによる導入方法では、シスコ製品からインターネット経由で Cisco.com（シスコ ライセンス サービス）に使用状況情報を直接送信します。導入に必要な追加コンポーネントはありません。

2. **HTTP プロキシによるダイレクト クラウド アクセス**：HTTP プロキシによるダイレクト クラウド アクセスの導入方法では、シスコ製品から、プロキシ サーバ（Smart Call Home Transport Gateway または市販のプロキシ（Apache など）のいずれか）を介して、インターネット経由で http://www.cisco.com のシスコ ライセンス サービスに使用状況情報を送信します。

3. **接続状態のオンプレミス コレクタを介した間接アクセス**：接続状態のオンプレミス コレクタを介した間接アクセスによる導入方法では、シスコ製品から、現地のライセンス認証局として機能するローカルに接続されたコレクタに使用状況情報を送信します。データベースの同期を保つため、周期的に情報が交換されます。

4. **非接続状態のオンプレミス コレクタを介した間接アクセス**：非接続状態のオンプレミス コレクタを介した間接アクセスによる導入方法では、シスコ製品から、現地のライセンス認証局として機能するローカルの接続が解除されたコレクタに使用状況情報を送信します。データベースの同期を保つため、不定期に（月に 1 回など）人による読み取りが可能な情報の交換が実施されます。

オプション **1** と **2** には簡単な導入オプションを、オプション **3** と **4** にはセキュアな環境展開オプションを提供します。オプション **3** および **4** には、スマート ソフトウェア サテライトによるサポートが提供されます。

シスコ製品とシスコ ライセンス サービスとの間の通信は、Smart Call Home ソフトウェアが対応します。Smart Call Home については、次を参照してください。 Call Home について （131 ページ）

Call Home について

Call Home では、重要なシステム ポリシーに関する電子メール ベースおよび http/https ベースの通知を提供します。ポケットベル サービスや XML ベースの自動化された解析アプリケーションとの互換性のために、さまざまなメッセージ フォーマットが用意されています。この機能を使用して、ネットワーク サポート エンジニアにポケットベルで連絡したり、ネットワーク オペレーション センターに電子メールを送信したりできます。また、Cisco Smart Call Home サービスを使用して TAC のケースを生成できます。Call Home 機能では、診断情報および環境の障害とイベントに関する情報が含まれるアラート メッセージを配信できます。

Call Home 機能では、複数の受信者（Call Home 宛先プロファイルと呼びます）にアラートを配信できます。各プロファイルには、設定可能なメッセージ フォーマットとコンテンツ カテゴリが含まれます。Cisco TAC へアラートを送信するための宛先が事前に定義されていますが、独自の宛先プロファイルを定義することもできます。メッセージを送信するように Call Home を設定すると、適切な CLI show コマンドが実行され、そのコマンドの出力がメッセージに添付されます。Call Home メッセージは次のフォーマットで配信されます。

- 1 または 2 行で障害を説明する、ポケットベルや印刷レポートに適したショート テキスト フォーマット。
- 詳細な情報を十分に書式が整えられたメッセージで提供する、ユーザが読むのに適したフル テキスト フォーマット。
- Extensible Markup Language（XML）と Adaptive Messaging Language（AML）XML schema definition（XSD）を使用する、コンピュータで読み取り可能な XML フォーマット。AML XSD は Cisco.com Web サイト（http://www.cisco.com/）で公開されています。XML フォーマットでは、シスコの TAC との通信が可能になります。

サポート対象の柔軟な消費モデル ライセンス

スマート ライセンスでは、柔軟な消費ライセンス モデルを使用します。このライセンス モデルは、設定されているポートの容量に基づきます。柔軟な消費モデル ライセンスをサポートするシャーシを購入する場合は、柔軟な消費モデル ライセンスを設定してライセンス機能を有効にする必要があります。柔軟な消費モデル ライセンスは、システムのすべてのポートの使用状況を毎日確認し、結果を Cisco.com の Smart Licensing Manager に報告します。

スマート ライセンスを使用したライセンスの設定

ハードウェアまたはソフトウェアで柔軟な消費モデル ライセンスを有効にするには、グローバル コンフィギュレーション モードでライセンスの **smart flexible-consumption enable** コマンドを使用します。ハードウェアまたはソフトウェアで柔軟な消費モデル ライセンスを無効にするには、グローバル コンフィギュレーション モードで **no license smart flexible-consumption enable** コマンドを使用します。

次の表に、NCS 560 のさまざまな柔軟な消費モデル ライセンスの消費パターンを示します。

表 *5:* 柔軟な消費ライセンス モデルの使用パターン

柔軟な消費モデル ライセンス	消費パターン
ESS-AC-100G-RTU-1	ルータに固定シャーシがある場合、ライセンスの消費チェックはシャーシで実行されます。
ADV-AC-100G-RTU-1	ルータに固定シャーシがある場合、ライセンスの消費チェックはシャーシで実行されます。

スマート ライセンスを使用したライセンスの設定

デバイスの登録とアクティブ化

スマート ライセンスのコンポーネントは、ncs560-mini-x.iso イメージにパッケージ化されます。Smart Call Home の設定に必要な HTTP クライアントは、ncs560-k9sec RPM にパッケージ化されます。ここで説明する手順を使用してデバイスを登録およびアクティブ化し、バーチャル アカウントに関連付けます。

デバイスを登録およびアクティブ化するには、次を行う必要があります。

- Cisco Smart Software Manager ポータルから、登録トークンを生成します。
- 登録トークンを使用して、CLI でデバイスを登録します。

ポータルから製品登録トークンを生成する

ライセンスを追加する製品を購入しておく必要があります。製品を購入すると、Cisco Smart Software Manager ポータルに対するユーザ名とパスワードが提供されます。このポータルから製品インスタンス登録トークンを生成できます。

1. スマート ソフトウェア ライセンシングで Cisco Smart software Manager にログインします。
2. [Inventory] メニューの下で、[General] タブをクリックします。
3. [New Token] をクリックして製品登録トークンを生成します。

   デバイスの登録およびアクティブ化に使用される新しいトークン値をコピーして、デバイスをバーチャル アカウントに関連付けます。

（注） このトークンは、290 日間有効です。

CLI での新しい製品の登録

CLI で、登録トークンを使用してデバイスをアクティブ化します。

```
RP/0/RP0/CPU0:router#license smart register idtoken token_ID
RP/0/RP0/CPU0:router#commit
```

登録が成功すると、デバイスでアイデンティティ証明書を受信します。この証明書はデバイスに保存され、それ以降のシスコとのすべての通信で自動的に使用されます。スマート ライセンスは、シスコへの登録情報を 290 日ごとに自動的に更新します。登録が失敗した場合、エラーがログに記録されます。また、ライセンスの使用状況データが収集され、毎月レポートが送信されます。必要に応じて、機密情報（ホスト名、ユーザ名、パスワードなど）が使用状況レポートから除外されるように、Smart Call Home 設定を構成できます。

スマート ライセンス設定の確認

スマート ライセンスを有効にした後、**show** コマンドを使用して、デフォルトのスマート ライセンス設定を確認できます。問題が検出された場合は、さらに設定を行う前に修正を行ってください。

手順

	コマンドまたはアクション	目的
ステップ 1	**show license status** 例： `RP/0/RP0/CPU0:router#show license status`	スマート ライセンスのコンプライアンス ステータスを表示します。以下は、表示される可能性があるステータスです。 • Waiting：デバイスがライセンス権限付与要求を行った後の初期状態を示します。デバイスはシスコとの通信を確立し、Cisco License Manager に正常に登録されます。 • Authorized：デバイスが Cisco License Manager と通信できること、およびライセンス権限付与の要求を開始する権限を持っていることを示します。 • Out-Of-Compliance：1 つ以上のライセンスがコンプライアンス違反であ

	コマンドまたはアクション	目的
		ることを示します。追加ライセンスを購入する必要があります。 • Eval Period：スマート ライセンスが評価期間が間もなく終了することを示します。Cisco Licensing Manager にデバイスを登録する必要があります。登録しない場合ライセンスが期限切れになります。 （注） スマート ライセンスの評価期限切れを警告するメッセージが 1 時間ごとに繰り返しコンソールに表示されますが、デバイスでは機能上の影響は見られません。これらの繰り返しメッセージを停止するには、新しい登録トークンを使用してデバイスを再度登録する必要があります。 • Disabled：スマート ライセンスが無効になっていることを示します。 • Invalid：データベースに存在しないため、シスコが権限付与タグを認識しないことを示します。
ステップ 2	**show license all** 例： `RP/0/RP0/CPU0:router#show license all`	使用中のすべての権限を表示します。さらに、関連付けられているライセンス証明書、コンプライアンスステータス、udi、およびその他の詳細が表示されます。
ステップ 3	**show license status** 例： `RP/0/RP0/CPU0:router#show license status`	使用中のすべての権限のステータスを表示します。
ステップ 4	**show license udi** 例： `RP/0/RP0/CPU0:router#show license udi`	UDI 情報を表示します。

	コマンドまたはアクション	目的
ステップ 5	**show license summary** 例： `RP/0/RP0/CPU0:router#show license summary`	使用中のすべての権限の概要を表示します。
ステップ 6	**show license platform summary** 例： `RP/0/RP0/CPU0:router#show license platform summary`	登録ステータスを表示し、一般的なライセンス モデルまたはボルテックス ライセンス モデルでの必須、詳細、およびトラッキング ライセンスの消費に関する詳細情報を提供します。
ステップ 7	**show license platform detail** 例： `RP/0/RP0/CPU0:router#show license platform detail`	一般的なモデルとボルテックス モデルの両方で、特定のプラットフォームで消費される可能性のあるライセンスの詳細を表示します。また、特定のライセンスの現在および次回の消費数も表示されます。アクティブなモデルの情報（ライセンス モデルが一般的かボルテックスか）を表示します。
ステップ 8	**show call-home smart-licensing statistics**	Smart Call Home を使用したスマート ライセンス マネージャとシスコ バックエンド間の通信について統計情報を表示します。通信が失敗したりドロップされる場合は、Call Home の設定に誤りがないか確認してください。

次に、**show call-home smart-licensing statistics** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router#show call-home smart-licensing statistics
Success: Successfully sent and response received.
Failed : Failed to send or response indicated error occurred.
Inqueue: In queue waiting to be sent.
Dropped: Dropped due to incorrect call-home configuration.

Msg Subtype      Success Failed  Inqueue Dropped Last-sent (GMT-07:00)
-------------------------------------------------------------------------
ENTITLEMENT      2       0       0       0       2014-04-24 18:24:34
REGISTRATION     1       0       0       0       2014-04-25 03:53:57
ACKNOWLEDGEMENT  1       0       0       0       2014-04-23 19:21:21
RENEW            1       0       0       0       2014-04-23 19:21:11
DEREGISTRATION   1       0       0       0       2014-04-25 03:31:35
```

スマート ライセンス登録の更新

一般に、登録は6ヵ月ごとに自動更新されます。登録をオンデマンドの手動更新にするには、このオプションを使用します。これにより、次回の登録更新サイクルまで6ヵ月待機する代わりに、このコマンドを実行してライセンスのステータスをすぐに確認できます。

始める前に

スマートライセンスを更新するには、次の条件が満たされていることを確認する必要があります。

- スマート ライセンスが有効になっている。
- デバイスが登録されている。

手順

	コマンドまたはアクション	目的

ステップ1	**license smart renew** {**auth** | **id**}
	例：
	`RP/0/RP0/CPU0:#license smart renew auth`
	
	`Tue Apr 22 09:12:37.086 PST`
	
	`license smart renew auth: Authorization process is in progress.`
	`Please check the syslog for the authorization status and result.`
		Cisco Smart Licensing を使用して ID または認証を更新します。ID 証明書の更新が失敗した場合は、製品インスタンスが未確認状態に移行し、評価期間の消費が開始されます。
		（注） 認証期間は、スマート ライセンス システムによって30日ごとに更新されます。ライセンスが「承認済み」または「コンプライアンス違反」（OOC）にある限り、認証期間が更新されます。猶予期間は、承認期間が過ぎると開始されます。猶予期間中、または猶予期間が「期限切れ」状態になると、システムは引き続き認証期間の更新を試行します。再試行に成功すると、新しい認証期間が開始されます。

スマート ライセンスの登録解除

デバイスがインベントリから移された場合、再導入のために別の場所に出荷された場合、または返品許可（RMA）プロセスを使用して交換のためにシスコに返送された場合は、登録解除オプションを使用してデバイスの登録をキャンセルできます。デバイスの登録をキャンセルするには、次の手順を実行します。

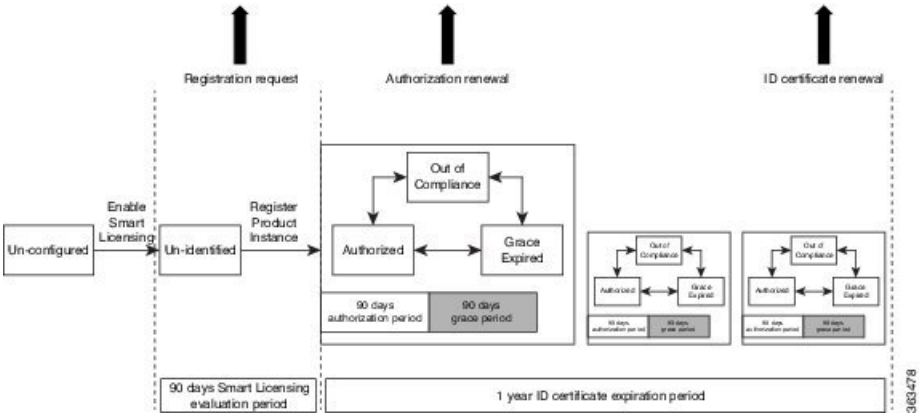
手順

	コマンドまたはアクション	目的

ステップ 1	**license smart deregister**
	例：
	`RP/0/RP0/CPU0 #license smart deregister`
	
	`license smart deregister: Success`
	
	`License command "license smart deregister " completed successfully.`
		デバイスの登録をキャンセルし、30 日間の評価モードに送信します。プラットフォームのすべてのスマート ライセンス資格と証明書が削除されます。
		（注） 製品インスタンスはシスコのライセンスクラウドサービスから登録解除されましたが、スマート ライセンスは引き続き有効になっています。

スマート ライセンスのワークフロー

スマート ライセンスのワークフローを、次のフローチャートに示します。



ライセンス、製品インスタンス、および登録トークン

ライセンス

ライセンスは、すべてのシスコ製品に必要です。シスコ製品のすべてのライセンスは、製品によって異なる次の 2 つのタイプのいずれかになります。

- 永久ライセンス：期限切れがないライセンス。
- 有効期限付きライセンス：一定期間（1 年、3 年、または購入した任意の期間）後に自動的に期限切れになるライセンス。

すべての製品ライセンスは、バーチャル アカウントに存在します。

製品インスタンス（Product Instances）

製品インスタンスとは、製品インスタンス登録トークン（または登録トークン）を使用して登録される、一意のデバイス ID（UDI）を使用した個々のデバイスです。1 つの登録トークンで 1 つの製品のインスタンスをいくつでも登録できます。各製品インスタンスで、同じバーチャルアカウント内に存在する 1 つ以上のライセンスを使用できます。製品インスタンスは、特定の更新期間中 Cisco Smart Software Manager サーバに定期的に接続する必要があります。製品インスタンスが接続に失敗するとライセンス不足であるとマークされますが、そのライセンスは引き続き使用されます。製品インスタンスを削除すると、そのライセンスがリリースされ、バーチャル アカウント内で使用可能になります。

製品インスタンスの登録トークン

製品は、登録するまで登録トークンが必要です。登録トークンは、エンタープライズアカウントに関連付けられた製品インスタンス登録トークンテーブルに格納されます。製品を登録すると登録トークンは不要になるため、そのテーブルから取り消したり、削除したりしても影響はありません。登録トークンの有効日数は、1 ～ 365 日間に設定できます。

仮想アカウント

スマート ライセンスでは、Smart Software Manager ポータル内で複数のライセンス プールまたはバーチャル アカウントを作成できます。[Virtual Accounts] オプションを使用して、組織の 1 つのセクションが別のセクションのライセンスを使用できないように、ライセンスをコストセンターに関連付けられた個別のバンドルに集約できます。たとえば、会社を地域で分ける場合、地域ごとにバーチャルアカウントを作成して、その地域のライセンスおよび製品インスタンスを保持できます。

注文プロセス中に別のライセンスを指定しない限り、新しいライセンスと製品インスタンスはすべて Smart Software Manager のデフォルトのバーチャル アカウントに配置されます。デフォルトのアカウントに配置されると、必要に応じて他のアカウントに転送できるようになりますが、これは必要なアクセス権限がある場合に限ります。

Smart Software Manager ポータル（https://tools.cisco.com/rhodui/index）を使用して、ライセンス プールを作成するか、ライセンスを転送します。

コンプライアンス レポート

スマート ライセンス契約の条件に規定されているとおり、インベントリとライセンスに関するコンプライアンス データが含まれたレポートが定期的に自動送信されます。これらのレポートには、次の 3 つのいずれかの形式が使用されます。

- **定期的な記録**：この記録は、特定の時点で保存された関連性のあるインベントリ データを使用して、定期的（設定可能）に生成されます。このレポートは、アーカイブ用として Cisco Cloud 内に保存されます。
- **手動による記録**：この記録は、特定の時点で保存された関連性のあるインベントリ データを使用して手動で生成できます。このレポートは、アーカイブ用として Cisco Cloud 内に保存されます。

- **コンプライアンス違反の警告レポート**：このレポートは、ライセンス コンプライアンスに関するイベントが発生したときに、自動または手動で生成されます。このレポートには完全なインベントリ データは含まれておらず、特定のソフトウェア ライセンスに対して不足している権限のみが記載されています。

  これらのレポートは、Smart Software Manager ポータル（https://tools.cisco.com/rhodui/index）から確認できます。

第 10 章

ゼロ タッチ プロビジョニングの設定

ゼロ タッチ プロビジョニング（ZTP）は、ルートスイッチ プロセッサ（RSP）およびルート プロセッサ（RP）のサード パーティ アプリケーション（TPA）として機能します。ZTP は、2 つの異なる操作を実行するように設計されました。

- 初期設定をダウンロードして適用します。
- シェル スクリプトをダウンロードして実行します。

ダウンロードしたファイルの内容が **!! IOS XR** で始まる場合はコンフィギュレーション ファイルと見なされ、ZTP がコンフィギュレーション ファイルで **apply_config** アクションを実行します。

ダウンロードしたファイルの内容が **#! /bin/bash**、 **#! /bin/sh**、または **#!/usr/bin/python** で始まる場合はスクリプト ファイルと見なされ、ZTP でスクリプトを実行します。

ZTP は次のように動作します。

1. 起動時に実行される XR スクリプトが DHCP 要求を呼び出します。
2. DHCP サーバがユーザ スクリプトまたは設定ファイルのいずれかを返します。
3. ユーザ スクリプトまたは設定ファイルをダウンロードします。
4. ダウンロードしたユーザ スクリプトを実行するか、またはダウンロードした設定を適用します。

ZTP プロセスでエラーが発生した場合、または ZTP が終了した場合は、ZTP プロセスを開始する前に存在した初期設定に戻します。

管理ポートとデータ ポート間の ZTP の切り替え

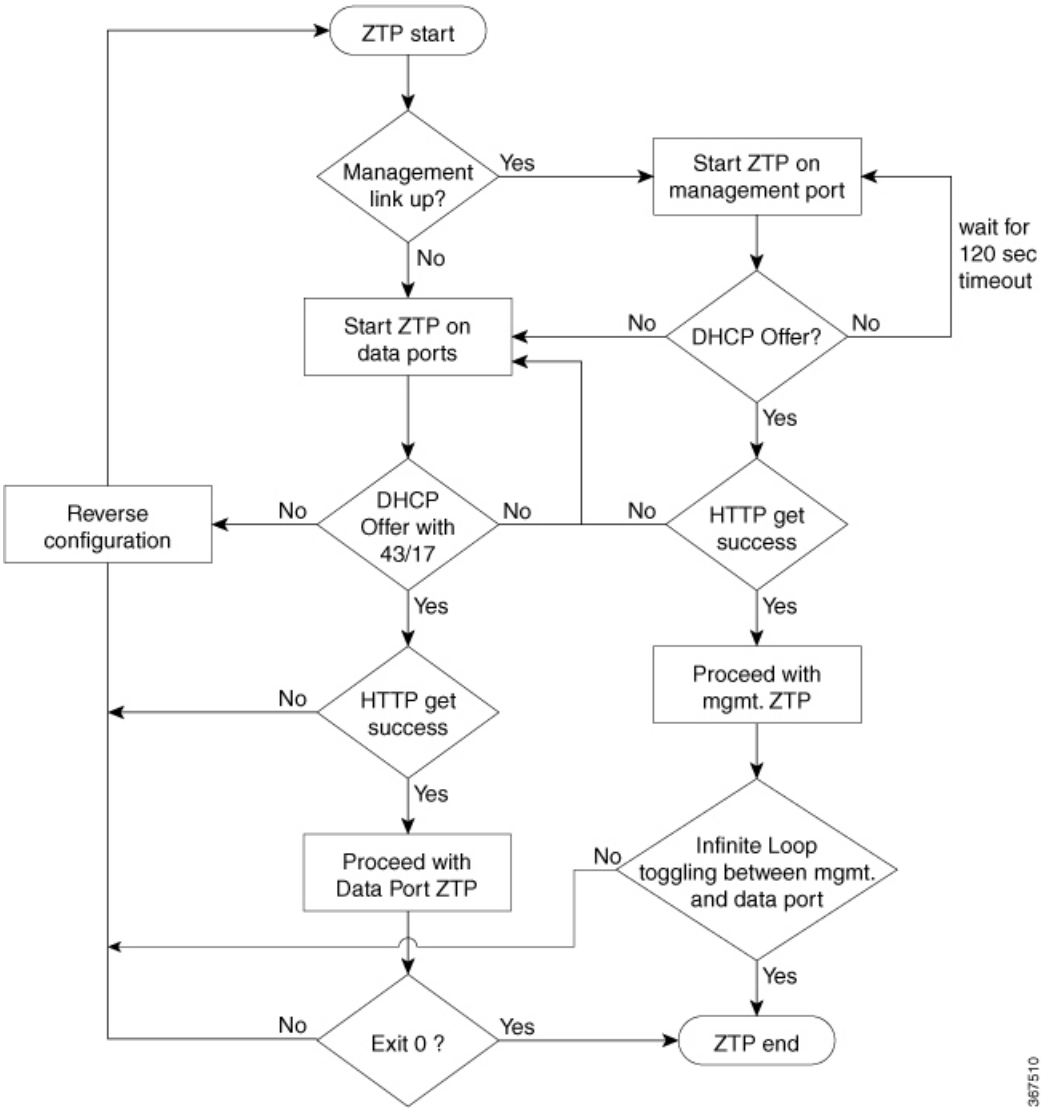
Cisco IOS XR リリース 6.5.1 から、ルータの新規起動時に、自動 ZTP プロセスが管理ポートから開始され、データ ポートに切り替わります。次の場合に、ZTPプロセスが管理ポートとデータ ポートの間で切り替えられます。

- ZTP が 10 秒以内にアクティブ インターフェイスを検出しない場合。
- ZTP が DHCP 応答を受信せず、dhclient 開始後の経過時間が 128 秒を超えている場合。

- ZTP でエラーが発生した場合。

以下のフロー図に、ZTP プロセスを示します。

図 *6 : ZTP* プロセス フロー シーケンス



（注）

- 新規起動時または手動での呼び出し中に、ZTP が dataport モードのすべてのデータ ポート インターフェイスで IPv6 をイネーブルにします。
- 自動ブレークアウト モードはサポートされていません。

この章は次のトピックで構成されています。

- 手動 ZTP 呼び出し （143 ページ）

• データ ポートでの認証 （144 ページ）
• ZTP ブートスクリプト （145 ページ）
• ZTP ユーティリティ （146 ページ）
• 例 （147 ページ）

手動 ZTP 呼び出し

手動ゼロ タッチプロビジョニング（ZTP）は、CLI コマンドを使用して手動で呼び出すことができます。この手動による方法は、ルータを段階的にプロビジョニングするのに役立ちます。再起動なしで ZTP 設定をテストするのに最適です。インターフェイス（データ ポートまたは管理ポート）上で ZTP を呼び出す場合は、最初にインターフェイスを起動して設定する必要はありません。 **ztp initiate** コマンドを実行すると、インターフェイスがダウンしていても、ZTP スクリプトでインターフェイスを起動し、dhclient が呼び出されます。したがって、ZTP はアップ状態かダウン状態かに関係なく、すべてのインターフェイスで実行できます。

ztp initiate、**ztp breakout**、**ztp terminate**、および **ztp clean** コマンドを使用して、ZTP をより多くのインターフェイス上で強制的に実行します。

- **ztp initiate**：新しい ZTP DHCP セッションを呼び出します。ログは、**/disk0:/ztp/ztp.log** にあります。
- **ztp terminate**：進行中の ZTP セッションを終了します。
- **ztp clean**：ディスクに保存されているすべての ZTP ファイルを削除します。

  リリース 6.2.3 以降、ログ ファイル ztp.log が **/var/log** フォルダに保存され、ログ ファイルのコピーがソフト リンクを使用して **/disk0:/ztp/ztp.log** で使用できるようになりました。ただし、**ztp clean** を実行すると、現在の ZTP ログが保存されている **/var/log** フォルダではなくディスクに保存されていたファイルがクリアされます。現在の ZTP からログを実行するには、**/var/log/** フォルダから ZTP ログ ファイルを手動でクリアする必要があります。

コマンドの詳細については、 の「ZTP command」の章を参照してください。

このタスクでは、手動 ZTP 呼び出しの最も一般的な使用例、4x10 ブレークアウト検出と ZTP の呼び出しを示しています。

手順

	コマンドまたはアクション	目的
ステップ 1	**ztp breakout** 例： `RP/0/RP0/CPU0:router# ztp breakout`	ブレイクアウトをサポートしている 100 GE インターフェイス上で 4x10 ブレークアウトを試行し、no-shut 後に動作がダウンします。10x10 ブレークアウト設定によって 10GE インターフェイスの動作が起動する場合は、ブレークアウト設定が維持されます。それ以外の場合は、設定が元に戻されます。

	コマンドまたはアクション	目的
ステップ 2	**ztp initiate dataport** 例： `RP/0/RP0/CPU0:router# ztp initiate dataport`	起動している、または起動可能なすべてのデータ ポートで DHCP セッションを呼び出します。ZTPはバックグラウンドで実行されます。進行状況を確認するには、**show logging** を使用するか、**/disk0:/ztp/ztp.log** を確認してください。

データ ポートでの認証

新規起動時は ZTP プロセスが管理ポートから開始されますが、データ ポートに切り替えることができます。DHCP サーバとの接続を検証するために、DHCP オプション 43（IPv4）、およびオプション 17（IPv6）を使用してデータ ポートで認証が実行されます。これらの DHCP オプションはオプション スペースで定義され、**dhcpd.conf** および **dhcpd6.conf** コンフィギュレーション ファイルに含まれています。オプション スペースを定義する際は、認証に次のパラメータを指定する必要があります。

- 認証コード：認証コードは 0 または 1 のいずれかです。0 は認証が不要であることを示し、1 は MD5 チェックサムが必要であることを示します。
- クライアント ID：クライアント ID は「exr-config」にする必要があります。
- MD5 チェックサム：これはシャーシのシリアル番号です。これは、**echo -n $SERIALNUMBER | md5sum | awk '{print $1}'** を使用して取得できます。

dhcpd.conf の設定例を次に示します。以下の例では、**VendorInfo** というオプション スペースが、認証用の 3 つのパラメータを使用して定義されています。

```
class "vendor-classes" {
   match option vendor-class-identifier;
}

option space VendorInfo;
option VendorInfo.clientId code 1 = string;
option VendorInfo.authCode code 2 = unsigned integer 8;
option VendorInfo.md5sum code 3 = string
option vendor-specific code 43 = encapsulate VendorInfo;
subnet 10.65.2.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option routers 10.65.2.1;
  range 10.65.2.1 10.65.2.200;
}
host xrv9k-1-mgmt {
   hardware ethernet 00:50:60:45:67:01;
   fixed-address 10.65.2.39;
   vendor-option-space VendorInfo;
   option VendorInfo.clientId "exr-config";
   option VendorInfo.authCode 1;
   option VendorInfo.md5sum "aedf5c457c36390c664f5942ac1ae3829";
   option bootfile-name "http://10.65.2.1:8800/admin-cmd.sh";
}
```

ZTP ブートスクリプト

ブートごとに実行するようにスクリプトをハードコードする場合は、次のように設定します。

```
conf t
    ztp bootscript /disk0:/myscript
commit
```

上記の設定では、最初のデータプレーンインターフェイスが設定されるまで待機し、その後、IPアドレスで管理インターフェイスが設定されるまでさらに1分待機することで、アプリケーションが使用するサードパーティの名前空間への接続が確実に行われます。遅延が望ましくない場合は、以下を使用します。

```
conf t
    ztp bootscript preip /disk0:/myscript
commit
```

（注） 上記のコマンドが最初に設定されると、今すぐ呼び出すかどうかを尋ねるプロンプトが表示されます。このプロンプトはテスト時に役立ちます。

次は **/disk0:/myscript** のコンテンツ例です。

```
#!/bin/bash
exec &> /dev/console # send logs to console
source /pkg/bin/ztp_helper.sh

# If we want to only run one time:
xrcmd "show running" | grep -q myhostname
if [[ $? -eq 0 ]]; then
    echo Already configured
fi

# Set the hostname
cat >/tmp/config <<%%
!! XR config example
hostname myhostname
%%
xrapply /tmp/config

#
# Force an invoke of ZTP again. If there was a username normally it would not run. This
 forces it.
# Kill off ztp if it is running already and suppress errors to the console when ztp runs
 below and
# cleans up xrcmd that invokes it. ztp will continue to run however.
#
xrcmd "ztp terminate noprompt" 2>/dev/null
xrcmd "ztp initiate noprompt" 2>/dev/null
```

ZTP ユーティリティ

ZTP には、ユーザ スクリプト内から送信可能な一連のシェル ユーティリティが含まれています。**ztp_helper.sh** は、ユーザ スクリプトによって送信可能なシェル スクリプトです。**ztp_helper.sh** は、一部の XR 機能にアクセスするための単純なユーティリティを提供します。以下は、呼び出すことができる bash 関数です。

- **xrcmd**：単一の XR exec コマンドを実行するために使用されます。

```
xrcmd “show running”
```

- **xrapply**：ファイルに指定された構成ブロックを適用します。

```
cat >/tmp/config <<%%
!! XR config example
hostname node1-mgmt-via-xrapply
%%
xrapply /tmp/config
```

- **xrapply_with_reason**：XR 構成のブロックをロギング目的の理由とともに適用するために使用されます。

```
cat >/tmp/config <<%%
!! XR config example
hostname node1-mgmt-via-xrapply
%%
xrapply_with_reason "this is a system upgrade" /tmp/config
```

- **xrapply_string**：XR 構成のブロックを 1 行で適用するために使用されます。

```
xrapply_string "hostname foo\ninterface GigabitEthernet0/0/0/0\nipv4 address 1.2.3.44
 255.255.255.0\n"
```

- **xrapply_string_with_reason**：ロギング目的の理由とともに XR 構成のブロックを 1 行で適用するために使用されます。

```
xrapply_string_with_reason ”system renamed again" "hostname venus\n interface
TenGigE0/0/0/0\n ipv4 address 172.30.0.144/24\n”
```

- **xrreplace**：XR 構成の置換をファイルを介して XR 名前空間に適用するために使用されます。

```
cat rtr.cfg <<%%
!! XR config example
hostname node1-mgmt-via-xrreplace
%%
xrreplace rtr.cfg
```

- **admincmd**：XR 名前空間で管理 CLI コマンドを実行するために使用されます。ログは、**/disk0:/ztp/ztp_admincmd.log** にあります。

```
admincmd running [show platform]
```

```
ztp-user connected from 192.0168.0.1 using console on host
sysadmin-vm:0_RP0# show platform | nomore
Tue Jan 30 23:12:30.757 UTC
Location Card Type HW State SW State Config State
------------------------------------------------------------------------------
0/RP0 NCS-5501 OPERATIONAL OPERATIONAL NSHUT
0/FT0 NCS-1RU-FAN-FW OPERATIONAL N/A NSHUT
0/FT1 NCS-1RU-FAN-FW OPERATIONAL N/A NSHUT
0/PM0 NCS-1100W-ACFW OPERATIONAL N/A NSHUT
0/PM1 NCS-1100W-ACFW OPERATIONAL N/A NSHUT
```

- **xrapply_with_extra_auth**：認証を必要とする XR 構成を、ファイルを介して XR 名前空間に適用するために使用されます。**xrapply_with_extra_auth** API が、エイリアス、フレックス グループなどの追加の認証を必要とする構成が適用されるときに使用されます。

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrapply_with_extra_auth >/tmp/config
```

- **xrreplace_with_extra_auth**：XR 構成の置換をファイルを介して XR 名前空間に適用するために使用されます。**xrreplace_with_extra_auth** API が、エイリアス、フレックス グループなどの追加の認証を必要とする構成が適用されるときに使用されます。

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrreplace_with_extra_auth >/tmp/config
```

例

ZFS はフラッシュ ファイル システムでの操作をディレクトリ **/disk0:/ztp/** に記録します。ZTP は、DHCP サーバとのすべてのトランザクションおよびすべての状態遷移を記録します。ZTP の先行実行も **/disk0:/ztp/old_logs/** に記録されます。

次の例に、コマンド **ztp initiate interface Ten 0/0/0/0 verbose** を使用してデータ インターフェイスからダウンロードした単純な構成スクリプトの実行を示します。このスクリプトは、システムのすべてのインターフェイスを開き、すべてのインターフェイスに 30 秒のロード間隔を設定します。

```
#!/bin/bash
##############################################################################
# *** Be careful this is powerful and can potentially destroy your system ***
#                *** !!! Use at your own risk !!! ***
#
# Script file should be saved on the backend HTTP server
##############################################################################
```

```
source ztp_helper.sh
config_file="/tmp/config.txt"
interfaces=$(xrcmd "show interfaces brief")

function activate_all_if(){
  arInt=($(echo $interfaces | grep -oE '(Te|Fo|Hu)[0-9]*/[0-9]*/[0-9]*/[0-9]*'))
  for int in ${arInt[*]}; do
    echo -ne "interface $int\n no shutdown\n load-interval 30\n" >> $config_file
  done
  xrapply_with_reason "Initial ZTP configuration" $config_file
}

### Script entry point
if [ -f $config_file ]; then
  /bin/rm -f $config_file
else
  /bin/touch $config_file
fi
activate_all_if;
exit 0
```

次に、ZTP ロギング出力の例を示します。

```
Oct 11 11:05:38 172.30.0.54 ztp-script: Hello from ncs-560-c !!!
Oct 11 11:05:40 172.30.0.54 ztp-script: current=6.1.1, desired=6.1.1
Oct 11 11:05:40 172.30.0.54 ztp-script: Version match, proceeding to configuration
Oct 11 11:05:41 172.30.0.54 ztp-script: Starting autoprovision process...
Oct 11 11:05:42 172.30.0.54 ztp-script: ### XR K9SEC INSTALL ###
Oct 11 11:05:44 172.30.0.54 ztp-script: ### Downloading complete ###
Oct 11 11:05:55 172.30.0.54 ztp-script: Waiting for k9sec package to be activated
Oct 11 11:06:01 172.30.0.54 ztp-script: ### XR K9SEC INSTALL COMPLETE ###
Oct 11 11:06:03 172.30.0.54 ztp-script: ### Installing midnight commander ###
Oct 11 11:06:04 172.30.0.54 ztp-script: ### Downloading system configuration ###
Oct 11 11:06:05 172.30.0.54 ztp-script: ### Downloading system configuration complete
###
Oct 11 11:06:06 172.30.0.54 ztp-script: ### Applying initial system configuration ###
Oct 11 11:06:11 172.30.0.54 ztp-script: !!! Checking for errors !!!
Oct 11 11:06:14 172.30.0.54 ztp-script: ### Applying system configuration complete ###
Oct 11 11:06:15 172.30.0.54 ztp-script: Autoprovision complete...
```