Cisco IOS XE Everest 16.6.x プログラマビリティ コンフィギュレーション ガイド

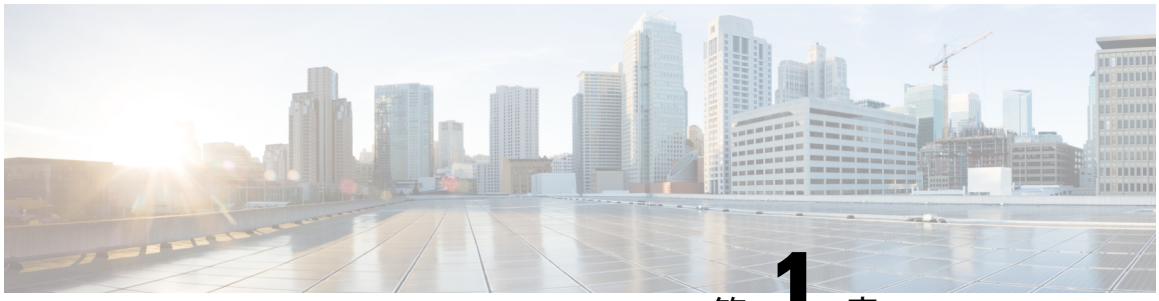
初版：2017 年 7 月 31 日

最終更新：2017 年 11 月 3 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/

© 2017 Cisco Systems, Inc. All rights reserved.

目 次



第 1 章

新機能および変更された機能に関する情報

この章では、すべての機能についてリリース固有の情報を記載しています。

• 新機能および変更機能に関する情報 （1 ページ）

新機能および変更機能に関する情報

次の表は、新機能および変更機能、サポート対象のプラットフォーム、および機能へのリンクをまとめたものです。

表 *1*：新機能および変更機能に関する情報

機能	リリースとプラットフォーム
プロビジョニング	
ゼロ タッチ プロビジョニング	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ

機能	リリースとプラットフォーム
ゼロ タッチ プロビジョニング：HTTP コピー	Cisco IOS XE Everest 16.6.1 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ
iPXE	Cisco IOS XE Denali 16.3.2 および Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3650 シリーズ スイッチ Cisco IOS XE Everest 16.6.1 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ
シェルとスクリプト化	
ゲスト シェル	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ

機能	リリースとプラットフォーム
Python API	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ
Python CLI モジュール	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ
EEM Python モジュール	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ

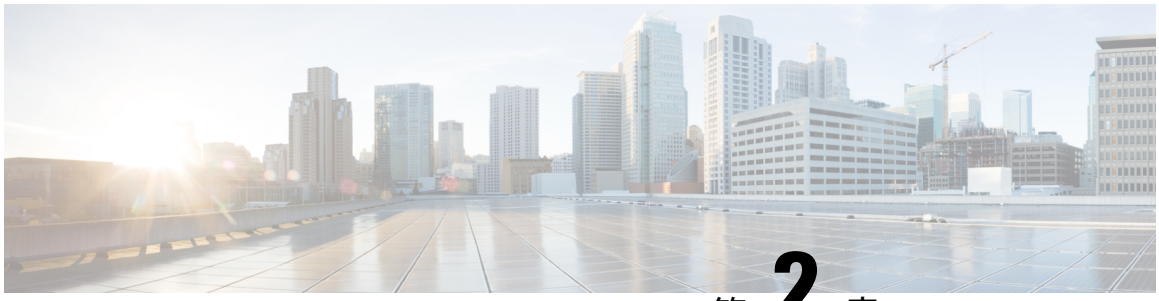
機能	リリースとプラットフォーム
モデル駆動型プログラマビリティ	
NETCONF ネットワーク管理インターフェイス	Cisco IOS XE Denali 16.3.1 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ
モデル駆動型テレメトリ NETCONF ダイヤルイン	Cisco IOS XE Everest 16.6.1 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ
サービス中モデル更新プログラム	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.1 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ

機能	リリースとプラットフォーム
RESTCONF ネットワーク管理インターフェイス	Cisco IOS XE Everest 16.6.1 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-HX および ASR1002-HX） • Cisco Cloud Services Router 1000V シリーズ

新機能および変更機能に関する情報

第 I 部

プロビジョニング



第 2 章

ゼロ タッチ プロビジョニング

ネットワーク プロビジョニングの課題に対応するため、シスコは、ゼロ タッチ プロビジョニング モデルを導入しました。このモジュールでは、ゼロ タッチ プロビジョニング機能について説明します。

（注） ゼロ タッチ プロビジョニング機能は自動的に有効になり、設定は不要です。

ゼロ タッチ プロビジョニング

ネットワーク プロビジョニングの課題に対応するため、シスコは、ゼロ タッチ プロビジョニング モデルを導入しました。このモジュールでは、ゼロ タッチ プロビジョニング機能について説明します。

（注） ゼロ タッチ プロビジョニング機能は自動的に有効になり、設定は不要です。

ゼロ タッチ プロビジョニングについて

ゼロ タッチ プロビジョニングの概要

ゼロ タッチプロビジョニングは、異機種混在ネットワーク環境でのネットワーク デバイス プロビジョニングを自動化する、オープン ブートストラップ インターフェイスを提供します。

ゼロ タッチプロビジョニングをサポートするデバイスが起動し、スタートアップ コンフィギュレーションが見つからない場合（初期インストール時）、デバイスはゼロ タッチ プロビジョニング モードに入ります。デバイスは、Dynamic Host Control Protocol（DHCP）サーバを検索し、インターフェイスの IP アドレス、ゲートウェイ、ドメイン ネーム システム（DNS）サーバの IP アドレスをブートストラップして、ゲスト シェルを有効にします。次にデバイスは

HTTP/TFTP サーバの IP アドレスまたは URL を取得し、HTTP/TFTP サーバからデバイスを構成する Python スクリプトをダウンロードします。

ゲスト シェルは、Python スクリプトを実行するための環境を提供します。ゲスト シェルは、ダウンロードした Python スクリプトを実行して、初期構成をデバイスに適用します。

初期プロビジョニングが完了したら、ゲスト シェルは有効化されたままになります。詳細については、「ゲスト シェル」の章を参照してください。

（注） ゼロ タッチ プロビジョニングが失敗した場合、デバイスは自動インストールにフォールバックして、コンフィギュレーション ファイルをロードします。詳細については、「Using AutoInstall and Setup」を参照してください。

ゼロ タッチ プロビジョニングのための DHCP サーバの設定

ゼロ タッチプロビジョニングでは、プロビジョニングされる新しいデバイスと同じネットワークで DHCP サーバを実行する必要があります。ゼロ タッチ プロビジョニングは、管理用ポートとインバンド ポートの両方でサポートされます。

新しいデバイスをオンにすると、そのデバイスは、Python スクリプトが存在する HTTP/TFTP サーバの IP アドレス情報と Python スクリプトのフォルダ パスを DHCP サーバから取得します。Python スクリプトの詳細については、「Python API」および「Python CLI モジュール」の各章を参照してください。

DHCP サーバは、次のオプションで DHCP 検出イベントに応答します。

- オプション 150：（任意）管理ネットワーク上の、実行される Python スクリプトをホストしている HTTP/TFTP サーバを指す IP アドレスの一覧が含まれます。
- オプション 67：HTTP/TFTP サーバ上の Python スクリプトのファイル パスが含まれます。

これらの DHCP オプションを受信すると、デバイスは、HTTP/TFTP サーバに接続して Python スクリプトをダウンロードします。この時点で、デバイスは HTTP/TFTP サーバに到達するルートを持たないため、DHCP サーバによって提供されるデフォルトのルートを使用します。

ゼロ タッチ プロビジョニングの構成例

TFTP コピーを使用しての管理ポートにおける DHCP サーバ設定の例

次に、デバイスの管理ポート経由で接続されている場合に TFTP コピーを使用して行う DHCP サーバ設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp excluded-address vrf Mgmt-vrf 10.1.1.1 10.1.1.10
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# vrf Mgmt-vrf
```

```
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 150 ip 203.0.113.254
Device(config-dhcp)# option 67 ascii /sample_python_dir/python_script.py
Device(config-dhcp)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# no ip dhcp client request tftp-server-address
Device(config-if)# end
```

HTTP コピーを使用しての管理ポートにおける DHCP サーバ設定の例

次に、デバイスの管理ポート経由で接続されている場合に HTTP コピーを使用して行う DHCP サーバ設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# vrf Mgmt-vrf
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 67 ascii http://198.51.100.1:8000/sample_python_2.py
Device(config-dhcp)# end
```

TFTP コピーを使用したインバンド ポートでのサンプル DHCP サーバ構成

次に示すのは、デバイスのインバンド ポート経由で接続されている場合の、TFTP コピーを使用したサンプル DHCP サーバ構成です。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 150 ip 203.0.113.254
Device(config-dhcp)# option 67 ascii /sample_python_dir/python_script.py
Device(config-dhcp)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# no ip dhcp client request tftp-server-address
Device(config-if)# end
```

HTTP コピーを使用したインバンド ポートでのサンプル DHCP サーバ構成

次に示すのは、デバイスのインバンド ポート経由で接続されている場合の、HTTP コピーを使用したサンプル DHCP サーバ構成です。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
```

```
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 67 ascii http://192.0.2.1:8000/sample_python_2.py
Device(config-dhcp)# end
```

Linux Ubuntu デバイス上でのサンプル DHCP サーバの構成

次の DHCP サーバ構成例は、サーバがデバイスの管理ポートまたはインバンド ポートのどちらかに接続されていることと、Python スクリプトが TFTP サーバからコピーされることを示しています。

```
root@ubuntu-server:/etc/dhcp# more dhcpd.conf
subnet 10.1.1.0 netmask 255.255.255.0 {
range 10.1.1.2  10.1.1.255;
      host 3850 {
          fixed-address                      10.1.1.246 ;
          hardware ethernet                  CC:D8:C1:85:6F:00;
          option bootfile-name !<opt 67>     " /python_dir/python_script.py";
         option tftp-server-name !<opt 150>  "203.0.113.254";
      }
  }
```

次のサンプル DHCP 構成は、Python スクリプトが HTTP サーバからデバイスにコピーされることを示しています。

```
Day0_with_mgmt_port_http
------------------------
subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.2 192.168.1.255;
      host C2-3850 {
          fixed-address                   192.168.1.246 ;
          hardware ethernet               CC:D8:C1:85:6F:00;
          option bootfile-name            "http://192.168.1.46/sample_python_2.py";
      }
  }
```

DHCP サーバが実行状態になったら、管理ネットワーク接続デバイスを起動します。これにより構成の残りの部分は自動的に実行されます。

サンプルの Python プロビジョニング スクリプト

次に示すのは、HTTP サーバまたは TFTP サーバのいずれかから使用できるサンプル Python スクリプトです。

```
print "\n\n *** Sample ZTP Day0 Python Script *** \n\n"

# Importing cli module
import cli


print "\n\n *** Executing show platform  *** \n\n"
cli_command = "show platform"
cli.executep(cli_command)
```

```
print "\n\n *** Executing show version *** \n\n"
cli_command = "show version"
cli.executep(cli_command)

print "\n\n *** Configuring a Loopback Interface *** \n\n"
cli.configurep(["interface loop 100", "ip address 10.10.10.10 255.255.255.255", "end"])


print "\n\n *** Executing show ip interface brief  *** \n\n"
cli_command = "sh ip int brief"
cli.executep(cli_command)

print "\n\n *** ZTP Day0 Python Script Execution Complete *** \n\n"
```

Cisco 4000 シリーズ サービス統合型ルータの起動ログ

次のゼロ タッチ プロビジョニングのブート ログでは、ゲスト シェルが正常に有効にされ、Python スクリプトがゲスト シェルにダウンロードされ、ゲスト シェルがダウンロードした Python スクリプトを実行してデバイスをデイ ゼロに設定していることが示されています。

```
% failed to initialize nvram
! <This message indicates that the startup configuration
is absent on the device. This is the first indication that the Day Zero work flow is
going to start.>

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco ISR4451-X/K9 (2RU) processor with 7941237K/6147K bytes of memory.
Processor board ID FJC1950D091
4 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
16777216K bytes of physical memory.
7341807K bytes of flash memory at bootflash:.
0K bytes of WebUI ODM Files at webui:.

%INIT: waited 0 seconds for NVRAM to be available


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: %
!!<DO NOT TOUCH. This is Zero-Touch Provisioning>>
Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)
```

```
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  down                  down
GigabitEthernet0/0/1   unassigned      YES unset  down                  down
GigabitEthernet0/0/2   unassigned      YES unset  down                  down
GigabitEthernet0/0/3   192.168.1.246   YES DHCP   up                    up
GigabitEthernet0       192.168.1.246   YES DHCP   up                    up
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** ZTP Day0 Python Script Execution Complete ***



Press RETURN to get started!
```

デイ ゼロ プロビジョニングが完了すると、IOS プロンプトがアクセス可能になります。

Cisco Catalyst 9000 シリーズ スイッチの起動ログ

次のセクションでは、ゼロタッチプロビジョニングの起動ログのサンプルを表示します。このようなログでは、ゲストシェルが正常に有効にされ、Python スクリプトがゲストシェルにダウンロードされ、ゲストシェルがダウンロードした Python スクリプトを実行してデバイスをデイゼロに設定していることが示されています。

```
% Checking backup nvram
% No config present. Using default config

FIPS: Flash Key Check : Begin
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled
```

```
! <This message indicates that the startup configuration
is absent on the device. This is the first indication that the Day Zero
 work flow is
going to start.>
```

Cisco IOS XE Everest 16.6.x から Cisco IOS XE Fuji 16.8.x へ

このセクションでは、.py スクリプトを実行する前の起動ログのサンプルを表示します。

```
Press RETURN to get started!

The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable


 *** Sample ZTP Day0 Python Script ***

...

 *** ZTP Day0 Python Script Execution Complete ***
```

このセクションでは、デイゼロプロビジョニング用にデバイスを設定する方法を示します。

```
Initializing Hardware...

System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Compiled Thu 02/20/2020 23:47:51.50 by rel

Current ROMMON image : Primary
Last reset cause     : SoftwareReload
C9300-48UXM platform with 8388608 Kbytes of main memory

Preparing to autoboot. [Press Ctrl-C to interrupt]  0
boot: attempting to boot from [flash:cat9k_iosxe.16.06.05.SPA.bin]
boot: reading file cat9k_iosxe.16.06.05.SPA.bin
########################################################################################################


Both links down, not waiting for other switches
Switch number is 1

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
```

```
Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE),
Version 16.6.5, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 10-Dec-18 12:52 by mcpre



Cisco IOS-XE software, Copyright (c) 2005-2018 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


% Checking backup nvram
% No config present. Using default config


FIPS: Flash Key Check : Begin
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco C9300-48UXM (X86) processor with 1392780K/6147K bytes of memory.
Processor board ID FCW2144L045
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045


%INIT: waited 0 seconds for NVRAM to be available

SETUP: new interface Vlan1 placed in "shutdown" state


Press RETURN to get started!
```

```
*Sep  4 20:35:07.330: %SMART_LIC-6-AGENT_READY: Smart Agent for Licensing is initialized
*Sep  4 20:35:07.493: %IOSXE_RP_NV-3-NV_ACCESS_FAIL: Initial read of NVRAM contents
failed
*Sep  4 20:35:07.551: %IOSXE_RP_NV-3-BACKUP_NV_ACCESS_FAIL: Initial read of backup NVRAM
 contents failed
*Sep  4 20:35:10.932: dev_pluggable_optics_selftest attribute table internally inconsistent
 @ 0x1D4

*Sep  4 20:35:13.406: %CRYPTO-4-AUDITWARN: Encryption audit check could not be performed
*Sep  4 20:35:13.480: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type vlan
*Sep  4 20:35:13.715: %LINK-3-UPDOWN: Interface Lsmpi18/3, changed state to up
*Sep  4 20:35:13.724: %LINK-3-UPDOWN: Interface EOBC18/1, changed state to up
*Sep  4 20:35:13.724: %LINEPROTO-5-UPDOWN: Line protocol on Interface LI-Null0, changed
 state to up
*Sep  4 20:35:13.724: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to
down
*Sep  4 20:35:13.725: %LINK-3-UPDOWN: Interface LIIN18/2, changed state to up
*Sep  4 20:35:13.749: WCM-PKI-SHIM: buffer allocation failed for SUDI support check
*Sep  4 20:35:13.749: PKI/SSL unable to send Sudi support to WCM
*Sep  4 20:35:14.622: %IOSXE_MGMTVRF-6-CREATE_SUCCESS_INFO: Management vrf Mgmt-vrf
created with ID 1,
    ipv4 table-id 0x1, ipv6 table-id 0x1E000001
*Sep  4 20:34:42.022: %STACKMGR-6-STACK_LINK_CHANGE: Switch 1 R0/0: stack_mgr:  Stack
port 1 on Switch 1 is nocable
*Sep  4 20:34:42.022: %STACKMGR-6-STACK_LINK_CHANGE: Switch 1 R0/0: stack_mgr:  Stack
port 2 on Switch 1 is down
*Sep  4 20:34:42.022: %STACKMGR-6-STACK_LINK_CHANGE: Switch 1 R0/0: stack_mgr:  Stack
port 2 on Switch 1 is nocable
*Sep  4 20:34:42.022: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr:  Switch 1 has
 been added to the stack.
*Sep  4 20:34:42.022: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr:  Switch 1 has
 been added to the stack.
*Sep  4 20:34:42.022: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr:  Switch 1 has
 been added to the stack.
*Sep  4 20:34:42.022: %STACKMGR-6-ACTIVE_ELECTED: Switch 1 R0/0: stack_mgr:  Switch 1
has been elected ACTIVE.
*Sep  4 20:35:14.728: %LINEPROTO-5-UPDOWN: Line protocol on Interface Lsmpi18/3, changed
 state to up
*Sep  4 20:35:14.728: %LINEPROTO-5-UPDOWN: Line protocol on Interface EOBC18/1, changed
 state to up
*Sep  4 20:35:15.506: %HMANRP-6-HMAN_IOS_CHANNEL_INFO: HMAN-IOS channel event for switch
 1: EMP_RELAY: Channel UP!
*Sep  4 20:35:15.510: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
state to down
*Sep  4 20:35:34.501: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively
 down
*Sep  4 20:35:34.717: %SYS-5-RESTART: System restarted --
Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.6.5,
 RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 10-Dec-18 12:52 by mcpre
*Sep  4 20:35:34.796: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Sep  4 20:35:35.266: %SYS-6-BOOTTIME: Time taken to reboot after reload =  283 seconds
*Sep  4 20:35:35.796: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0,
 changed state to up
*Sep  4 20:35:36.607: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/1, changed state to
down
*Sep  4 20:35:36.607: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/2, changed state to
down
*Sep  4 20:35:36.607: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/3, changed state to
down
```

```
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/4, changed state to
down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/1, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/2, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/3, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/4, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/5, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/6, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/7, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/8, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface FortyGigabitEthernet1/1/1, changed state
 to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface FortyGigabitEthernet1/1/2, changed state
 to down
*Sep  4 20:35:37.607: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/1,
 changed state to down
*Sep  4 20:35:37.608: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/2,
 changed state to down
*Sep  4 20:35:37.608: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/3,
 changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/4,
 changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/1, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/2, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/3, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/4, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/5, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/6, changed state to down
*Sep  4 20:35:43.511: AUTOINSTALL: Obtain tftp server address (opt 150) 159.14.27.2
*Sep  4 20:35:43.511: PNPA: Setting autoinstall complete to true for 159.14.27.2
*Sep  4 20:35:57.673: %PLATFORM_PM-6-FRULINK_INSERTED: 8x10G uplink module inserted in
the switch 1 slot 1
*Sep  4 20:36:19.562: [IOX DEBUG] Guestshell start API is being invoked

*Sep  4 20:36:19.562: [IOX DEBUG] provided idb is mgmt interface

*Sep  4 20:36:19.562: [IOX DEBUG] Setting up guestshell to use mgmt-intf

*Sep  4 20:36:19.562: [IOX DEBUG] Setting up chasfs for iox related activity

*Sep  4 20:36:19.562: [IOX DEBUG] Setting up for iox pre-clean activity if needed

*Sep  4 20:36:19.562: [IOX DEBUG] Waiting for iox pre-clean setup to take affect

*Sep  4 20:36:19.562: [IOX DEBUG] Waited for 1 sec(s) for iox pre-clean setup to take
affect

*Sep  4 20:36:19.562: [IOX DEBUG] Auto-configuring iox feature

*Sep  4 20:36:19.563: [IOX DEBUG] Waiting for CAF and ioxman to be up, in that order
```

```
*Sep  4 20:36:20.076: %UICFGEXP-6-SERVER_NOTIFIED_START: Switch 1 R0/0: psd:  Server iox
 has been notified to start
*Sep  4 20:36:23.564: [IOX DEBUG] Waiting for another 5 secs

*Sep  4 20:36:28.564: [IOX DEBUG] Waiting for another 5 secs
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:33.564: [IOX DEBUG] Waiting for another 5 secs
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:34.564: [IOX DEBUG] Waited for 16 sec(s) for CAF and ioxman to come up

*Sep  4 20:36:34.564: [IOX DEBUG] Validating if CAF and ioxman are running

*Sep  4 20:36:34.564: [IOX DEBUG] CAF and ioxman are up and running

*Sep  4 20:36:34.564: [IOX DEBUG] Building the simple mgmt-intf enable command string

*Sep  4 20:36:34.564: [IOX DEBUG] Enable command is: request platform software iox-manager

    app-hosting guestshell enable

*Sep  4 20:36:34.564: [IOX DEBUG] Issuing guestshell enable command and waiting for it
to be up
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:38.578: [IOX DEBUG] Waiting for another 5 secs
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:39.416: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/0/48, changed state
 to up
*Sep  4 20:36:40.416: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/0/48,
    changed state to upThe process for the command is not responding or is otherwise
unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:43.586: [IOX DEBUG] Waiting for another 5 secs
Guestshell enabled successfully

*Sep  4 20:37:45.321: [IOX DEBUG] Checking for guestshell mount path

*Sep  4 20:37:45.321: [IOX DEBUG] Validating if guestshell is ready for use

*Sep  4 20:37:45.321: [IOX DEBUG] Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***



 *** Executing show platform  ***

Switch  Ports    Model               Serial No.   MAC address     Hw Ver.       Sw Ver.

------  -----   ---------           -----------  --------------  -------      --------
```

```
 1       62     C9300-48UXM           FCW2144L045  ec1d.8b0a.6800  V01           16.6.5

Switch/Stack Mac Address : ec1d.8b0a.6800 - Local Mac Address
Mac persistency wait time: Indefinite
                                     Current
Switch#   Role        Priority      State
-----------------------------------------
*1       Active          1          Ready


 *** Executing show version ***


Cisco IOS XE Software, Version 16.06.05
Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.6.5,
 RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 10-Dec-18 12:52 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2018 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
ROM: IOS-XE ROMMON
BOOTLDR: System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Switch uptime is 2 minutes
Uptime for this control processor is 4 minutes
System returned to ROM by Reload Command
System image file is "flash:cat9k_iosxe.16.06.05.SPA.bin"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Technology Package License Information:
------------------------------------------------------------------
Technology-package                  Technology-package
Current             Type            Next reboot
------------------------------------------------------------------
network-advantage   Permanent       network-advantage
cisco C9300-48UXM (X86) processor with 1392780K/6147K bytes of memory.
Processor board ID FCW2144L045
36 Ethernet interfaces
1 Virtual Ethernet interface
4 Gigabit Ethernet interfaces
20 Ten Gigabit Ethernet interfaces
2 Forty Gigabit Ethernet interfaces
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
```

```
0K bytes of WebUI ODM Files at webui:.
Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045
Switch Ports Model              SW Version        SW Image              Mode
------ ----- -----              ----------        ----------            ----
*    1 62    C9300-48UXM        16.6.5            CAT9K_IOSXE           BUNDLE
Configuration register is 0x102


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES unset  administratively down down
GigabitEthernet0/0     10.127.128.3    YES DHCP   up                    up
Tw1/0/1                unassigned      YES unset  down                  down
Tw1/0/2                unassigned      YES unset  down                  down
Tw1/0/3                unassigned      YES unset  down                  down
Tw1/0/4                unassigned      YES unset  down                  down
Tw1/0/5                unassigned      YES unset  down                  down
Tw1/0/6                unassigned      YES unset  down                  down
Tw1/0/7                unassigned      YES unset  down                  down
Tw1/0/8                unassigned      YES unset  down                  down
Tw1/0/9                unassigned      YES unset  down                  down
Tw1/0/10               unassigned      YES unset  down                  down
Tw1/0/11               unassigned      YES unset  down                  down
Tw1/0/12               unassigned      YES unset  down                  down
Tw1/0/13               unassigned      YES unset  down                  down
Tw1/0/14               unassigned      YES unset  down                  down
Tw1/0/15               unassigned      YES unset  down                  down
Tw1/0/16               unassigned      YES unset  down                  down
Tw1/0/17               unassigned      YES unset  down                  down
Tw1/0/18               unassigned      YES unset  down                  down
Tw1/0/19               unassigned      YES unset  down                  down
Tw1/0/20               unassigned      YES unset  down                  down
Tw1/0/21               unassigned      YES unset  down                  down
Tw1/0/22               unassigned      YES unset  down                  down
Tw1/0/23               unassigned      YES unset  down                  down
Tw1/0/24               unassigned      YES unset  down                  down
Tw1/0/25               unassigned      YES unset  down                  down
Tw1/0/26               unassigned      YES unset  down                  down
Tw1/0/27               unassigned      YES unset  down                  down
Tw1/0/28               unassigned      YES unset  down                  down
Tw1/0/29               unassigned      YES unset  down                  down
Tw1/0/30               unassigned      YES unset  down                  down
Tw1/0/31               unassigned      YES unset  down                  down
Tw1/0/32               unassigned      YES unset  down                  down
Tw1/0/33               unassigned      YES unset  down                  down
Tw1/0/34               unassigned      YES unset  down                  down
Tw1/0/35               unassigned      YES unset  down                  down
Tw1/0/36               unassigned      YES unset  down                  down
```

```
Te1/0/37               unassigned      YES unset  down                  down
Te1/0/38               unassigned      YES unset  down                  down
Te1/0/39               unassigned      YES unset  down                  down
Te1/0/40               unassigned      YES unset  down                  down
Te1/0/41               unassigned      YES unset  down                  down
Te1/0/42               unassigned      YES unset  down                  down
Te1/0/43               unassigned      YES unset  down                  down
Te1/0/44               unassigned      YES unset  down                  down
Te1/0/45               unassigned      YES unset  down                  down
Te1/0/46               unassigned      YES unset  down                  down
Te1/0/47               unassigned      YES unset  down                  down
Te1/0/48               unassigned      YES unset  up                    up
GigabitEthernet1/1/1   unassigned      YES unset  down                  down
GigabitEthernet1/1/2   unassigned      YES unset  down                  down
GigabitEthernet1/1/3   unassigned      YES unset  down                  down
GigabitEthernet1/1/4   unassigned      YES unset  down                  down
Te1/1/1                unassigned      YES unset  down                  down
Te1/1/2                unassigned      YES unset  down                  down
Te1/1/3                unassigned      YES unset  down                  down
Te1/1/4                unassigned      YES unset  down                  down
Te1/1/5                unassigned      YES unset  down                  down
Te1/1/6                unassigned      YES unset  down                  down
Te1/1/7                unassigned      YES unset  down                  down
Te1/1/8                unassigned      YES unset  down                  down
Fo1/1/1                unassigned      YES unset  down                  down
Fo1/1/2                unassigned      YES unset  down                  down
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** Configuring username, password, SSH  ***


Line 1 SUCCESS: username cisco privilege 15 password cisco
Line 2 SUCCESS: ip domain name domain
Line 3 SUCCESS: line vty 0 15
Line 4 SUCCESS: login local
Line 5 SUCCESS: transport input all
Line 6 SUCCESS: end


 *** ZTP Day0 Python Script Execution Complete ***
```

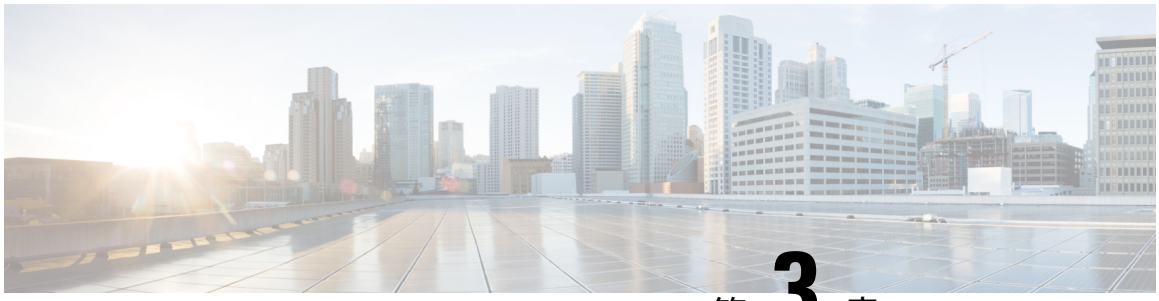
ゼロ タッチ プロビジョニングの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *2:* ゼロ タッチ プロビジョニングの機能情報

機能名	リリース	機能情報
ゼロ タッチ プロビジョニング	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	ネットワーク プロビジョニングの課題に対応するため、シスコは、ゼロ タッチ プロビジョニング モデルを導入しました。 Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • ゲストシェルをサポートするための、最低 8 GB の RAM を搭載した Cisco 4000 シリーズ サービス統合型ルータモデル。
ゼロ タッチ プロビジョニング：HTTP ダウンロード	Cisco IOS XE Everest 16.6.1	ゼロ タッチプロビジョニングは、HTTP および TFTP のファイル ダウンロードをサポートします。 Cisco IOS XE Everest 16.6.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 3 章

iPXE

iPXE は、ネットワーク ブーティングのオープン スタンダードである Pre-boot eXecution Environment（PXE）の拡張版です。このモジュールでは、iPXE 機能および設定方法について説明します。

iPXE について

iPXE について

iPXE は、ネットワーク ブーティングのオープン スタンダードである Pre-boot eXecution Environment（PXE）の拡張版です。

iPXE ネットブートは、次を提供します。

- IPv4 および IPv6 プロトコル
- FTP/HTTP/TFTP ブート イメージのダウンロード
- イメージへの埋め込みスクリプト
- Dynamic Host Configuration Protocol バージョン 6（DHCPv6）のためのステートレス アドレス自動設定（SLAAC）およびステートフル IP 自動設定のバリエーション、ブート URI、および IPv6 ルータ アドバタイズメントに応じた DHCPv6 オプションのパラメータ。

（注） Catalyst 9000 シリーズ スイッチでは、IPv6 はサポートされていません。

ネットブート要件

ネット ブーティングの主な要件は、次のとおりです。

- 適切に設定された DHCP サーバ。
- FTP/HTTP/TFTP サーバ上で使用可能なブート イメージ。
- ネットワーク ベースのソースから起動するように設定されたデバイス。

iPXE の概要

ネットワーク ブートローダは、ネットワーク ベースのソースからのブート処理をサポートします。ブートローダは、HTTP、FTP、またはTFTPサーバにあるイメージを起動します。ネットワーク ブート ソースは、iPXE のようなソリューションを使用して自動検出されます。

iPXE により、オフラインのデバイスのネットワーク ブートが可能になります。iPXE ブート モードには、次の 3 つのタイプがあります。

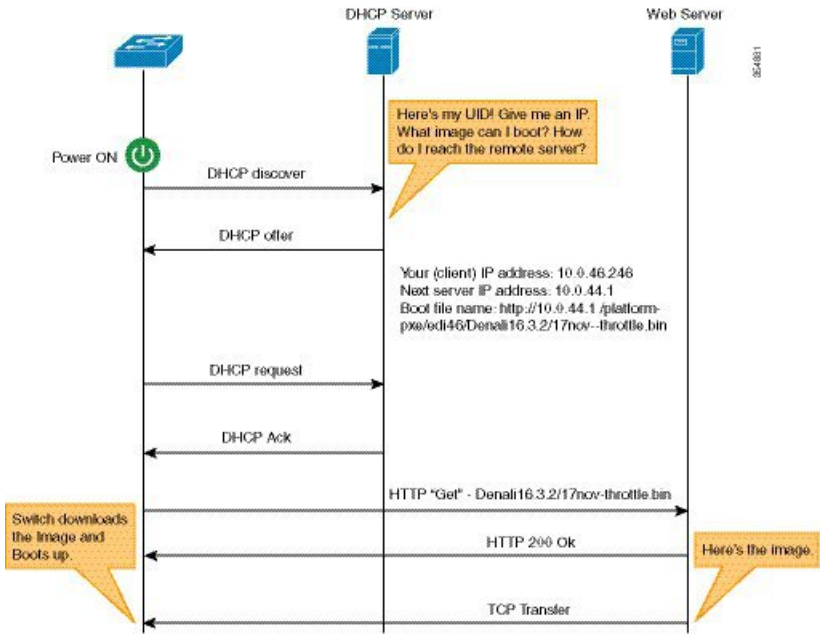
- iPXE タイムアウト：iPXE ネットワーク ブートを介して起動します。IPXE_TIMEOUT ROMmon 変数を使用して、iPXE ネットワーク ブートのタイムアウトを秒単位で設定します。iPXE タイムアウトを設定するには **boot ipxe timeout** コマンドを使用します。タイムアウト時間を経過すると、デバイス ブートがアクティブになります。
- iPXE 期限なし：iPXE ネットワーク ブートを介して起動します。**boot ipxe forever** コマンドが設定されている場合、デバイスは DHCP 要求を期限なしで送信します。これは iPXE のみを使うブートです（つまり、ブートローダは、有効なDHCP応答を受け取るまでDHCP要求を期限なしで送信するため、デバイス ブートまたはコマンド プロンプトにフォールバックすることはありません）。
- デバイス：設定されているローカル デバイスの BOOT 行を使ってブートします。デバイス ブートが設定された場合、設定されている IPXE_TIMEOUT ROMmon 変数は無視されます。デバイス ブートは、デフォルトのブート モードです。

（注） このマニュアルでは、手動ブートという用語も使われています。手動ブートは、ROMmon のリロードを行うかどうかを決定するフラグです。デバイスが ROMmon モードの場合は、手動で **boot** コマンドを実行する必要があります。

手動ブートが 1 に設定されている場合、ROMmon またはデバイス プロンプトがアクティブになります。手動ブートが0に設定されている場合、デバイスはリロードされますが、ROMmon モードはアクティブになりません。

ここでは、iPXE ブートローダの動作について説明します。

図 *1 : iPXE* ブートローダのワークフロー



1. ブートローダが DHCP 要求を送信します。
2. DHCP 応答には、IP アドレスとのブート ファイル名が含まれています。ブート ファイル名は、ブート イメージが TFTP サーバ（tftp://server/filename）、FTP サーバ（ftp://userid:password@server/filename）、または HTTP サーバ（http://server/filename）から取得されることを示しています。現在の iPXE 実装は管理ポート（GigabitEthernet0/0）のみを経由して動作するため、前面パネル ポートを介して送信される DHCP 要求はサポートされていません。
3. ブートローダがネットワーク ソースからイメージをダウンロードして起動します。
4. DHCP 応答が受信されない場合、ブートローダはブート モードの設定に基づいて、DHCP 要求を期限なしで、または指定された期間の間送信し続けます。タイムアウトが発生すると、ブートローダはデバイスベースのブートに戻ります。設定されたブート モードが **ipxe-forever** の場合のみ、デバイスは DHCP 要求を期限なしで送信します。**ipxe-timeout** ブート モード コマンドが設定されている場合、DHCP 要求は指定された時間にわたって送信され、タイムアウトが経過すると、デバイス ブート モードがアクティブになります。

手動ブートが無効になっている場合、ブートローダは、設定された ROMmon iPXE 変数の値に基づいて、デバイス ブートを実行するかネットワーク ブートを実行するかを決定します。手動ブートが有効か無効かにかかわらず、ブートローダは BOOTMODE 変数を使用して、デバイス ブートとネットワーク ブートのどちらを実行するかを決定します。手動ブートは、ユーザによって **boot manual switch** コマンドが設定済みであることを意味します。手動ブートが無効になっている場合にデバイスをリロードすると、起動プロセスが自動的に開始されます。

iPXE が無効になっている場合は、デバイスの起動方法の決定に、既存の BOOT 変数の内容が使用されます。BOOT 変数には、ネットワークベースの Uniform Resource Identifier（URI）（たとえば、http://、ftp://、tftp://）が含まれている場合があり、ネットワーク ブートが開始されま

す。しかし、ネットワーク イメージ パスの取得に DHCP は使用されません。デバイス IP アドレスは、IP_ADDRESS 変数から取得されます。BOOT 変数には、デバイスのファイル システム ベースのパスが含まれている場合もあり、この場合は、デバイスのファイル システム ベースのブートが開始されます。

起動に使用される DHCP サーバは、製品 ID（PID）（DHCP オプション 60 で判別可能）、シャーシのシリアル番号（DHCP オプション 61 で判別可能）、またはデバイスの MAC アドレスを使用して、デバイスを識別できます。**show inventory** および **show switch** コマンドでもデバイスでこれらの値を表示します。

次に、**show inventory** コマンドの出力例を示します。

```
Device# show inventory

NAME:"c38xx Stack", DESCR:"c38xx Stack"
PID:WS-3850-12X-48U-L, VID:V01 ,  SN: F0C1911V01A

NAME:"Switch 1", DESCR:"WS-C3850-12X48U-L"
PID:WS-C3850-12X48U-L, VID:V01 , SN:F0C1911V01A

NAME:"Switch1 -Power Supply B", DESCR:"Switch1 -Power Supply B"
PID:PWR-C1-1100WAC, VID:V01, SN:LIT1847146Q
```

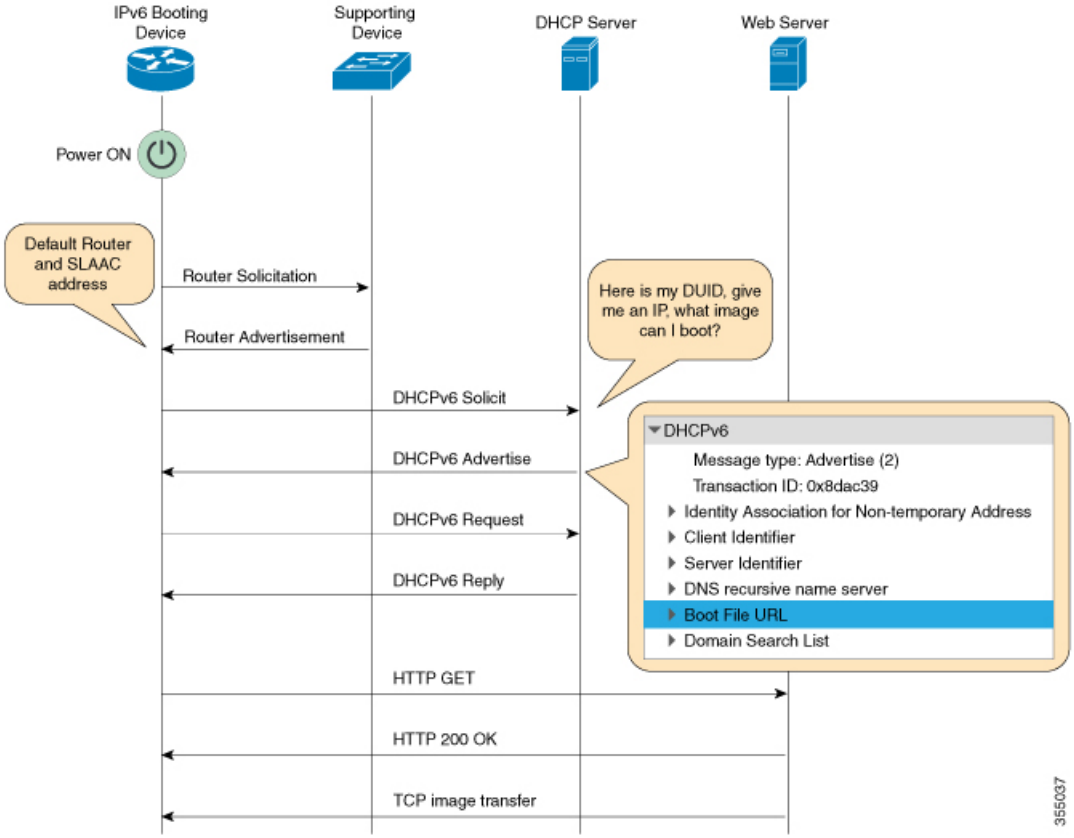
次の ROMmon 変数が iPXE に設定されている必要があります。

- BOOTMODE = ipxe-forever | ipxe-timeout | device
- IPXE_TIMEOUT = seconds

IPv6 iPXE ネットワーク ブート

Catalyst 9000 シリーズ スイッチでは、IPv6 はサポートされていません。

次の図は、Cisco デバイス上の IPv6 iPXE ネットワーク ブートの動作を表します。



次に、上掲の図の 4 つの要素を説明します。

- IPv6 ブート デバイス：iPXE ブートによって起動するデバイス。
- サポート デバイス：IPv6 アドレスで、ルータ アドバタイズメント（RA）メッセージを生成するように設定された Cisco デバイス。

（注） この図では、IPv6 ブート デバイス、サポート デバイス、および DHCP サーバは、同じサブネット上にあります。ただし、サポート デバイスと DHCP サーバが異なるサブネット上にある場合、ネットワーク内にリレーエージェントを設ける必要があります。

- DHCP サーバ：任意のオープン ソースの DHCP サーバ。
- Web サーバ：任意のオープン ソースの Web サーバ。

この項では、IPv6 iPXE ブート プロセスを説明します。

1. デバイスは、ルータ要請である Internet Control Message Protocol IPv6（ICMPv6）タイプ 133 パケットをローカル サブネット上の IPv6 デバイスに送信します。

2. ローカル サブネット上の IPv6 デバイスは、RA である ICMPv6 タイプ 134 パケットで応答します。ルータ要請メッセージを送信したデバイスは、ステートレス アドレス自動設定（SLAAC）アドレスを完成させるため、RA パケットからデフォルト ルータとプレフィックスの情報を取得します。

3. デバイスは、DHCP バージョン 6（DHCPv6）要請メッセージを、すべての DHCP エージェントについて、マルチキャスト グループ アドレス ff02::1:2 に送信します。

   次に、iPXE ブートの際の DHCPv6 要請パケットのフィールドの例を示します。

```
DHCPv6
 Message type: Solicit (1)
 Transaction ID: 0x36f5f1
 Client Identifier
 Vendor Class
 Identity Association for Non-Temporary Address
 Option Request
 User Class
 Vendor-specific Information
```

   DHCPv6 要請メッセージには、次の情報が含まれています。

   - DHCP 固有識別子（DUID）：クライアントを識別します。iPXE では、DUID-EN をサポートしています。EN は、エンタープライズ番号（Enterprise Number）の略です。この DUID は、ベンダーに割り当てられた固有の識別子に基づいています。
   - DHCPv6 オプション 3
   - DHCPv6 オプション 6
   - DHCPv6 オプション 15
   - DHCPv6 オプション 16
   - DHCPv6 オプション 17

4. DHCPv6 サーバが設定されている場合、そのサーバは、128 ビット IPv6 アドレス、ブートファイルの Uniform Resource Identifier（URI）、ドメイン ネーム システム（DNS）サーバおよびドメイン検索リスト、ならびにクライアントとサーバの ID を含む DHCPv6 アドバタイズ パケットで応答します。クライアント ID にはクライアント（この図では IPv6 ブート デバイス）の DUID が、サーバ ID には DHCPv6 サーバの DUID が、それぞれ含まれています。

5. それを受け、クライアントは、マルチキャスト グループ アドレス ff02::1:2 に DHCPv6 要求パケットを送信し、アドバタイズされたパラメータを要求します。

6. サーバは、クライアントのリンク ローカル（FE80::）の IPv6 アドレスにユニキャスト DHCPv6 応答を返します。次に、DHCPv6 応答パケットのフィールドの例を示します。

```
DHCPv6
 Message type: Reply (7)
 Transaction ID: 0x790950
 Identity Association for Non-Temporary Address
 Client Identifier
```

```
Server Identifier
DNS recursive name server
Boot File URL
Domain Search List
```

7. 次に、デバイスは、Web サーバに HTTP GET 要求を送信します。
8. 要求されたイメージが指定されたパスで使用可能な場合、Web サーバは、HTTP GET 要求に OK を返します。
9. TCP イメージ転送によりイメージがコピーされ、デバイスが起動します。

ROMmon モードでの IPv6 アドレスの割り当て

（注） Catalyst 9000 シリーズ スイッチでは、IPv6 はサポートされていません。

DHCP クライアントは、次の優先順位を使用して、ROMmon モードで使用する IPv6 アドレスを決定します。

1. DHCP サーバによって割り当てられたアドレス
2. ステートレス アドレス自動設定（SLAAC）アドレス
3. リンクローカル アドレス
4. サイトローカル アドレス

デバイスは、イメージをブートするのに DHCP サーバによって割り当てられたアドレスを使用します。DHCPv6 サーバがアドレスの割り当てに失敗した場合、デバイスは、SLAAC アドレスの使用を試行します。DHCP サーバによって割り当てられたアドレスと SLAAC アドレスの両方が使用できない場合、デバイスは、リンクローカル アドレスを使用します。ただし、イメージのコピーを正常に行うには、リモート FTP/HTTP/TFTP サーバがデバイスと同じローカル サブネット上にある必要があります。

最初の 3 つのアドレスが使用できない場合、デバイスは、自動的に生成されるサイトローカル アドレスを使用します。

iPXE がサポートする DHCP オプション

iPXE ブートは、ROMmon モードで次の DHCPv4 および DHCPv6 オプションをサポートしています。

（注） DHCP オプション 77 以外のオプションは、Catalyst 9000 シリーズ スイッチではサポートされていません。

- DHCP オプション 77：ユーザ クラス オプション。このオプションは、DHCP 検出パケットに追加されるもので、iPXE という文字列に等しい値を含んでいます。このオプションは、DHCP サーバからブートするためのイメージを探す iPXE DHCP クライアントを分離する際に使用されます。

  次に、ISC DHCP サーバからの DHCPv4 設定で、オプション 77 の使用が示されている例を示します。この例における if 条件は、オプション 77 が存在しており、文字列 iPXE に等しい場合は、イメージのブート ファイルの URI がアドバタイズされることを示します。

```
host Switch2 {
     fixed-address 192.168.1.20 ;
     hardware ethernet CC:D8:C1:85:6F:11 ;
           #user-class = length of string + ASCII code for iPXE
     if exists user-class and option user-class = 04:68:50:58:45 {
         filename "http://192.168.1.146/test-image.bin"
     }
}
```

- DHCPv6 オプション 15：ユーザ クラス オプション。このオプションは、DHCPv6 要請メッセージ内の IPv6 ユーザ クラス オプションです。次に、ISC DHCP サーバで定義されているオプション 15 の例を示します。

```
option dhcp6.user-class code 15 = string ;
```

  次に、DHCPv6 オプション 15 が使用されている DHCP サーバ設定の例を示します。

```
#Client-specific parameters
host switch1 {
     #assigning a fixed IPv6 address
     fixed-address6 2001:DB8::CAFE ;
     #Client DUID in hexadecimal format contains: DUID-type"2" + "EN=9" + "Chassis
serial number"
    host-identifier option dhcp6.client-id      00:02:00:00:00:09:46:4F:43:31:38:33:
31:58:31:41:53;
    #User class 00:04:69:50:58:45 is len 4 + "iPXE"
    if option dhcp6.user-class = 00:04:69:50:58:45 {
       option dhcp6.bootfile-url
"http://[2001:DB8::461/platform-pxe/edi46/test-image.bin";
    }
}
```

- DHCPv6 オプション 16：ベンダー クラス オプション。デバイスの製品 ID（PID）が含まれています。PID は、**show inventory** コマンドの出力または MODEL_NUM ROMmon 変数から特定できます。オプション 16 は ISC DHCP サーバのデフォルトのオプションではなく、次のように定義することができます。

```
option dhcp6.vendor-class-data code 16 = string;
```

  次に、DHCPv6 オプション 16 が使用されている設定例を示します。

```
# Source: dhcpd6ConfigPD

host host1-ipxe6-auto-host1 {
     fixed-address6 2001:DB8::1234;
```

```
     host-identifier option dhcp6.client-id 00:02:00:00:00:09:46:4F:
     43:31:38:33:31:58:31:41:53;
     if option dhcp6.vendor-class-data = 00:00:00:09:00:0E:57:53:2D:
     43:33:38:35:30:2D:32:34:50:2D:4D {
     option dhcp6.bootfile-url
"http://[2001:DB8::46]/platform-pxe/host1/17jan-polaris.bin";
```

次の表で、この出力に表示される重要なフィールドを説明します。

表 *3:* サンプル出力フィールドの説明

フィールド	説明
dhcp6.client-id	クライアントを識別する DHCP 固有識別子（DUID）。
dhcp6.user-class	DHCPv6 オプション 15、ユーザ クラス オプション。
dhcp6.vendor-class-data	DHCPv6 オプション 16、スイッチの製品 ID（PID）を含むベンダークラス オプション。
該当なし	一時的でないアドレスを要求する DHCPv6 オプション 3。
該当なし	DHCPv6 オプション 17、シスコに割り当てられているエンタープライズ ID *9* を含むベンダー識別オプション。
dhcp6.bootfile-url	ブート ファイル URI を要求する DHCPv6 オプション 6。

DHCPv6 固有識別子

Catalyst 9000 シリーズ スイッチでは、IPv6 はサポートされていません。

RFC 3315 によって定義されている DHCPv6 識別子（DUID）には、次の 3 種類があります。

- DUID-LLT：DUID リンク層アドレスと時刻。DHCP デバイスに接続しているネットワーク インターフェイスのリンク層アドレスに、生成された時刻のタイムスタンプが追加されたものです。
- DUID-EN：EN は、エンタープライズ番号（Enterprise Number）の略です。この DUID は、ベンダーに割り当てられた固有の ID に基づいています。
- DUID-LL：DHCP（クライアント/サーバ）デバイスに永久的に接続されているネットワーク インターフェイスのリンク層アドレスを使用して形成される DUID です。

シスコデバイスは、DHCP クライアント（DHCPv6 要請パケット内のデバイス）を識別するのに DUID EN（DUID タイプ 2）を使用します。

iPXE の設定方法

iPXE の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	• **boot ipxe forever** *switch number*
	• **boot ipxe timeout** *seconds switch number*
	例：
	`Device(config)# boot ipxe forever switch 2`
	例：
	`Device(config)# boot ipxe timeout 30 switch 2`
		BOOTMODE ROMmon 変数を設定します。
		• **forever** キーワードは、BOOTMODE ROMmon 変数を IPXE-FOREVER として設定します。
		• **timeout** キーワードは、BOOTMODE ROMmon 変数を IPXE-TIMEOUT として設定します。

ステップ 4	**boot system** {**switch** *switch-number* | **all**} {**flash:** | **ftp:** | **http:** | **tftp:**}
	例：
	`Device(config)# boot system switch 1 http://192.0.2.42/image-filename`
	または
	`Device(config)# boot system switch 1 http://[2001:db8::1]/image-filename`
		指定した場所からイメージを起動します。
		• リモートの FTP/HTTP/TFTP サーバには、IPv4 または IPv6 アドレスを使用できます。
		• 角かっこ内に IPv6 アドレスを入力する必要があります（RFC 2732 に従って）。そうしない場合、デバイスは起動しません。
		（注） Catalyst 9000 シリーズ スイッチでは、IPv6 はサポートされていません。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

デバイス ブートの設定

デバイス ブートは、**no boot ipxe** または **default boot ipxe** コマンドのいずれかを使用して設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	• **no boot ipxe**
	• **default boot ipxe**
	例：
	`Device(config)# no boot ipxe`
	例：
	`Device(config)# default boot ipxe`
		デバイス ブートを設定します。デフォルトのブート モードはデバイス ブートです。
		デバイスでデフォルト設定を有効にします。

ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

iPXE の設定例

例：iPXE 構成

以下は、デバイスがイメージで起動するまで、DHCP 要求を期限なしで送信するように iPXE を設定する例を示しています。

```
Device# configure terminal
Device(config)# boot ipxe forever switch 2
Device(config)# end
```

以下は、ブート モードを ipxe-timeout に設定する方法の例を示します。設定されているタイムアウト値は 200 秒です。設定されているタイムアウト経過後に iPXE ブート障害が発生する場合、設定されているデバイス ブートがアクティブになります。この例で、設定済みのデバイス ブートは http://[2001:db8::1]/image-filename です。

```
Device# configure terminal
Device(config)# boot ipxe timeout 200 switch 2
Device(config)# boot system http://[2001:db8::1]/image-filename
Device(config)# end
```

（注） Catalyst 9000 シリーズ スイッチでは、IPv6 はサポートされていません。

サンプルの iPXE ブート ログ

次に示すのは、ROMmon モードのデバイスからのサンプル ブート ログです。ここでは、**ipxe-timeout** コマンドを使用した手動ブートが設定されます。

```
switch: boot

pxemode:(ipxe-timeout) 60s timeout
00267.887 ipxe_get_booturl: Get URL from DHCP; timeout 60s
00267.953 ipxe_get_booturl: trying DHCPv6 (#1) for 10s
IPv4:
           ip addr 192.168.1.246
           netmask 255.255.255.0
           gateway 192.168.1.46
IPv6:
link-local addr fe80::ced8:c1ff:fe85:6f00
site-local addr fec0::ced8:c1ff:fe85:6f00
      DHCP addr 2001:db8::cafe
    router addr fe80::f29e:63ff:fe42:4756
     SLAAC addr 2001:db8::ced8:c1ff:fe85:6f00 /64
Common:
        macaddr cc:d8:c1:85:6f:00
            dns 2001:db8::46
       bootfile
http://[2001:DB8::461/platform-pxe/edi46/17jan-dev.bin--13103--2017-Feb28--13-54-50
         domain cisco.com
00269.321 ipxe_get_booturl: got URL
(http://[2001:DB8::461/platform-pxe/edi46/17jan-dev.bin--13103--2017-Feb-28--13-54-50)
Reading full image into memory ….…….…….…….…….…….…….…….…….…….…….…….…….…….…….…….…….…..
Bundle Image
---------------------------------------------
Kernel Address    : 0x5377a7e4
Kernel Size       : 0x365e3c/3563068
Initramfs Address : 0x53ae0620
Initramfs Size    : 0x13a76f0/20608752
```

```
Compression Format: mzip
```

iPXE 用のサンプル DHCPv6 サーバ構成

次に示すのは、参照用に ISC DHCP サーバから取られた DHCPv6 サーバ構成例です。先頭に文字 # がある行は、続く構成を説明しているコメントです。

```
Default-least-time 600;
max-lease-time-7200;
log-facility local7;

#Global configuration
#domain search list
option dhcp6.domain-search "cisco.com" ;
#User-defined options:new-name code new-code = definition ;
option dhcp6.user-class code 15 = string ;
option dhcp6.vendor-class-data code 16 = string;

subnet6 2001:db8::/64 {
 #subnet range for clients requiring an address
 range6 2001:db8:0000:0000::/64;

#DNS server options
option dhcp6.name-servers 2001:db8::46;

}
#Client-specific parameters
host switch1 {
    #assigning a fixed IPv6 address
    fixed-address6 2001:DB8::CAFE ;
    #Client DUID in hexadecimal that contains: DUID-type "2" + "EN=9" + "Chassis serial
 number"
    host-identifier option dhcp6.client-id 00:02:00:00:00:09:46:4F:43:31:38:33:
31:58:31:41:53;
   option dhcp6.bootfile-url "http://[2001:DB8::461/platform-pxe/edi46/test-image.bin";
}
```

DHCP サーバ コマンドの詳細については、ISC DHCP サーバ の Web サイトを参照してください。

この設定例では、dhcp6.client-id オプションはスイッチを識別し、エンタープライズ クライアント DUID が続きます。クライアント DUID は、16 進形式の 00:02 + 00:00:00:09 + のシャーシ シリアル番号を理解するために分解できます。ここで 2 はエンタープライズ クライアント DUID タイプ、9 はシスコのエンタープライズ DUID の予約済みコードをそれぞれ参照し、16 進形式でのシャーシ シリアル番号の ASCII コードが続きます。このサンプルのスイッチのシャーシ シリアル番号は、FOC1831X1AS です。

ブート ファイル URI は、指定された DUID を使用してのみスイッチにアドバタイズされます。

DHCPv6 ベンダー クラス オプション 16 も、DHCP サーバ上のスイッチを識別するため使用できます。デフォルトでは、この DHCP オプションは ISC DHCP サーバによっ

てサポートされていません。それをユーザ定義のオプションとして定義するには、次のように設定します。

```
option dhcp6.vendor-class-data code 16 = string;
```

次に示すのは、スイッチ製品 ID を使用して形成された DHCPv6 ベンダー クラス オプション 16 に基づいてスイッチを識別する、DHCP サーバの構成例です。

```
# Source: dhcp6ConfigPID

host edi-46-ipxe6-auto-edi46 {
    fixed-address6 2001:DB8::1234;
    host-identifier option dhcp6.client-id 00:02:00:00:00:09:
    46:4F:43:31:38:33:31:58:31:58:31:41:53;
    if option dhcp6.vendor-class-data = 00:00:00:09:00:0E:57:
    53:2D:43:33:38:35:30:2D:32:34:50:2D:4C {
      option dhcp6.bootfile-url "http://[2001:DB8::461/platform-pxe/edi46/17jan-dev.bin";

    }
}
```

この構成例では、dhcp6.vendor-class-data オプションは、DHCPv6 オプション 16 を参照します。dhcp6.vendor-class-data で、00:00:00:09 はシスコのエンタープライズ DUID、0E は PID の長さ、および残りは 16 進形式の PID です。PID は、**show inventory** コマンドまたは CFG_MODEL_NUM ROMmon 変数の出力から特定することもできます。このサンプル構成で使用される PID は、WS-C3850-24P-L です。

サーバ構成の DHCPv6 オプションおよび DUID は、ISC DHCP サーバのガイドラインに従って、16 進形式で指定する必要があります。

iPXE のトラブルシューティングのヒント

この項では、トラブルシューティングのヒントを説明します。

- 電源投入時に iPXE ブートが有効化されると、デバイスは、最初に DHCPv6 要請メッセージの送信を試行し、その後で、DHCPv4 検出メッセージの送信を試行します。ブート モードが **ipxe-forever** の場合、デバイスは、この 2 つを期限なしで反復し続けます。
- 起動モードが iPXE タイムアウトの場合、デバイスは、最初に DHCPv6 要請メッセージを、次に DHCPv4 検出メッセージを送信した後、タイムアウト時間が経過すると、デバイス ブートにフォールバックします。
- iPXE ブートを中断するには、コンソールにシリアル ブレイクを送信します。

  UNIX Telnet クライアントを使用している場合は、Ctrl キーを押した状態で] キーを押すと、ブレイクが送信されます。その他の Telnet クライアントを使用している場合、またはシリアル ポートに直接接続している場合は、ブレイクの送信のトリガーは、別のキーストロークまたはコマンドの場合があります。

- DHCP サーバはイメージで応答するものの DNS サーバがホスト名を解決できない場合、DNS デバッグを有効にします。
- HTTP サーバの接続をテストするには、HTTP コピーを使用して、HTTP サーバから少量のサンプル ファイルをデバイスにコピーします。たとえば ROMmon プロンプトで、**copy http://192.168.1.1/test null:**（フラッシュが通常はロックされており、テストに Null デバイスを使用する必要がある場合）または **http://[2001:db8::99]/test** と入力します。
- 手動ブートが有効化されており、ブート モードが iPXE タイムアウトである場合、デバイスが電源投入時に自動的に起動することはありません。ROMmon モードで **boot** コマンドを実行します。ブート プロセスが電源投入時に自動で発生するようにするには、手動ブートを無効にします。
- ROMmon モード の IPv6 アドレスやデフォルト ルータを含む現在の IPv6 パラメータを表示するには、**net6-show** コマンドを使用します。
- 設定に基づいて、**net-dhcp** または **net6-dhcp** コマンドを使用します。**net-dhcp** コマンドは DHCPv4 用のテスト コマンド、**net6-dhcp** コマンドは DHCPv6 用のテスト コマンドです。
- 名前を解決するには、**dig** コマンドを使用します。
- Web サーバからの HTTP 応答コードを表示するには、HTTP デバッグ ログを有効にします。
- SLAAC アドレスが生成されない場合、IPv6 RA メッセージを提供するルータがありません。この場合、IPv6 での iPXE ブートは、リンクローカルまたはサイトローカルのアドレスでのみ使用できます。

iPXE に関する追加情報

関連資料

関連項目	マニュアル タイトル
プログラマビリティ コマンド	『Programmability Command Reference, Cisco IOS XE Everest 16.6.1』

標準および RFC

標準/RFC	タイトル
RFC 3315	『*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*』
RFC 3986	『*Uniform Resource Identifier (URI): Generic Syntax*』

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

iPXE の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

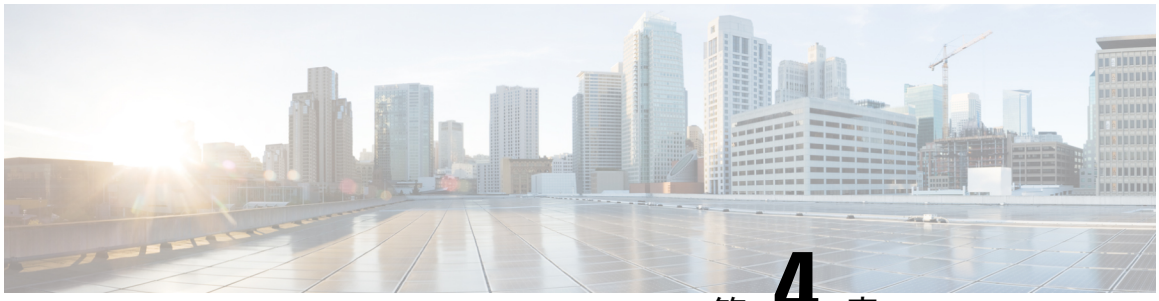
プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *4 : iPXE* の機能情報

機能名	リリース	機能情報
iPXE	Cisco IOS XE Denali 16.5.1a	ネットワーク ブートローダは、IPv4/IPv6 デバイス ベースまたはネットワーク ベースの送信元からのブート処理をサポートします。ネットワーク ブート ソースは、iPXE のようなソリューションを使用して自動的に検出される必要があります。 この機能は、次のプラットフォームに実装されていました。 • Catalyst 3650 シリーズ スイッチ • Catalyst 3850 シリーズ スイッチ
	Cisco IOS XE Denali 16.6.1	iPXE IPv6 は Catalyst 9000 シリーズ スイッチではサポートされていません。 この機能は、次のプラットフォームに実装されていました。 • Catalyst 9300 シリーズ スイッチ • Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。

第 II 部

シェルとスクリプト化



第 4 章

ゲスト シェル

ゲスト シェルは仮想化された Linux ベースの環境で、Python などの カスタム Linux アプリケーションを実行して Cisco デバイスを自動で制御および管理するために設計されています。システムの自動プロビジョニング（デイ ゼロ）も含まれます。このコンテナ シェルは、ホスト デバイスから分離された安全な環境を提供します。ユーザはそこで、スクリプトまたはソフトウェア パッケージをインストールし、実行することができます。

このモジュールでは、ゲスト シェルとそれを有効にする方法について説明します。

ゲスト シェルについて

ゲスト シェルの概要

ゲスト シェルは仮想化された Linux ベースの環境で、Python などの カスタム Linux アプリケーションを実行して Cisco デバイスを自動で制御および管理するために設計されています。ゲスト シェルを使用して、サード パーティ製 Linux アプリケーションをインストール、更新、および操作することもできます。システムイメージとともにバンドルされており、**guestshell enable** コマンドを使用してインストールできます。

ゲスト シェル環境は、ネットワーキングではなく、ツール、Linux ユーティリティ、および管理性を意図したものです。

ゲスト シェルは、ホスト（Cisco スイッチおよびルータ）システムとカーネルを共有します。ユーザは、ゲスト シェルの Linux シェルにアクセスし、コンテナの rootfs にあるスクリプトおよびソフトウェア パッケージを更新することができます。ただし、ゲスト シェル内のユーザは、ホストのファイル システムおよびプロセスを変更することはできません。

ゲスト シェル コンテナは、IOx を使用して管理されます。IOx は、Cisco IOS XE デバイスのためのシスコのアプリケーション ホスティング インフラストラクチャです。IOx は、シスコ、パートナー、およびサードパーティの開発者によって開発されたアプリケーションおよびサービスをネットワーク エッジ デバイスでシームレスにホスティングすることを、各種の多様なハードウェア プラットフォームにおいて可能にします。

次の表は、ゲスト シェルのさまざまな機能とサポート対象のプラットフォームに関する情報を提供します。

表 *5 : Cisco* ゲスト シェルの機能

	ゲスト シェル Lite（限定的な LXC コンテナ）	ゲスト シェル（LXC コンテナ）
オペレーティング システム	Cisco IOS XE	Cisco IOS XE
サポートされるプラットフォーム	• Cisco Catalyst 3650 シリーズ スイッチ（全モデル） • Cisco Catalyst 3850 シリーズ スイッチ（全モデル）	• Cisco ISR 4000 シリーズ サービス統合型ルータ（最低 8 GB の RAM を有するモデル）
ゲスト シェル環境	Montavista CGE7	CentOS 7
Python 2.7	サポート対象（Python V2.7.11）	サポート対象（Python V2.7.5）
カスタムの Python ライブラリ	• Cisco 組込イベント マネージャ • Cisco IOS XE CLI • Ncclient	• Cisco 組込イベント マネージャ • Cisco IOS XE CLI
サポートされる rootfs	Busybox、SSH、および Python PIP のインストール	SSH、Yum のインストール、および Python PIP のインストール
GNU C コンパイラ	サポート対象外	サポート対象外
RPM のインストール	サポート対象外	サポートあり
アーキテクチャ	MIPS	x86

ゲスト シェルとゲスト シェル Lite

ゲストシェル コンテナを使用すると、ユーザは、システム上で自分のスクリプトやアプリケーションを実行できるようになります。Intel x86 プラットフォーム上のゲスト シェル コンテナは、CentOS 7.0 の最小限の rootfs を持つ Linux コンテナ（LXC）になります。ランタイム中に、CentOS 7.0 で Yum ユーティリティを使用して、Python バージョン 3.0 などの他の Python ライ

ブラリをインストールすることができます。また、PIP を使用して Python パッケージをインストールまたは更新することもできます。

Catalyst 3650 や Catalyst 3850 シリーズ スイッチなどの MIPS プラットフォーム上のゲスト シェル Lite コンテナには、Carrier Grade Edition（CGE）7.0 の rootfs があります。ゲスト シェル Lite では、スクリプトのインストールまたは実行のみ可能です。これらのデバイスでは、Yum のインストールはサポートされていません。

ゲスト シェルのセキュリティ

シスコは、ゲスト シェル内のユーザまたはアプリケーションによってホスト システムが攻撃されることがないよう、セキュリティを提供しています。ゲスト シェルは、ホスト カーネルから分離され、非特権コンテナとして動作します。

ゲスト シェルのハードウェア要件

この項では、サポート対象のプラットフォームにおけるハードウェア要件に関する情報を提供します。

表 *6 : Catalyst* スイッチでのゲスト シェルのサポート

プラットフォーム	デフォルトの **DRAM**	ゲスト シェルのサポート
WS-3650-xxx（すべて）	4 GB	サポート対象
WS-3850-xxx（すべて）	4 GB	サポート対象
C9300-xx-x（すべて）	8 GB	サポート対象
C9500-24Q-x（すべて）	16 GB	サポート対象

Catalyst 3850 シリーズ スイッチの最小システム要件は、4 GB の DRAM です。

表 *7 : ISR 4000* シリーズ サービス統合型ルータでのゲスト シェルのサポート

プラットフォーム	デフォルトの **DRAM**	ゲスト シェルのサポート
ISR 4221	4GB	未サポート
ISR 4321	4 GB	未サポート
	8 GB	サポート対象
ISR 4331	8 GB	サポート対象
	16 GB	サポート対象
ISR 4351	8 GB	サポート対象
	16 GB	サポート対象

プラットフォーム	デフォルトの **DRAM**	ゲスト シェルのサポート
ISR 4431	8 GB	サポート対象
	16 GB	サポート対象
ISR 4451	8 GB	サポート対象
	16 GB	サポート対象

ISR 4000 シリーズ サービス統合型ルータの最小システム要件は、8 GB の DRAM です。

（注） 仮想サービスがインストールされているアプリケーションとゲスト シェル コンテナを同時に使用することはできません。

ゲスト シェルのストレージ要件

Catalyst 3650 および Catalyst 3850 シリーズ スイッチでは、ゲスト シェルは、フラッシュのファイル システムにのみインストールできます。Catalyst 3850 シリーズ スイッチのブートフラッシュでは、ゲスト シェルを正常にインストールするには 75 MB のディスク空き容量が必要です。

Cisco 4000 シリーズ サービス統合型ルータでは、ゲスト シェルは、ネットワーク インターフェイス モジュール（NIM）のサービス セット識別子（SSID）（ハード ディスク）がある場合、そこにインストールされます。ハード ディスク ドライブが使用可能な場合、ゲスト シェルのインストールにブートフラッシュを選択することはできません。Cisco 4000 シリーズ サービス統合型ルータでは、ゲスト シェルを正常にインストールするには 1100 MB のハード ディスク（NIM SSID）空き容量が必要です。

ゲスト シェルのインストール中にハード ディスク容量が不足した場合、エラー メッセージが表示されます。

次に、ISR 4000 シリーズ ルータでのエラー メッセージの例を示します。

```
% Error:guestshell_setup.sh returned error:255, message:
Not enough storage for installing guestshell. Need 1100 MB free space.
```

ブートフラッシュまたはハード ディスクの空き領域は、ゲスト シェルが追加データを格納するために使用されることがあります。Cisco Catalyst 3850 シリーズ スイッチでは、ゲスト シェルが使用できるストレージ 容量は 18 MB です。Cisco 4000 シリーズ サービス統合型ルータでは、ゲスト シェルが使用できるストレージ容量は 800 MB です。ゲスト シェルはブートフラッシュにアクセスするため、その空き領域の全体を使用できます。

表 *8 :* ゲスト シェルおよびゲスト シェル *Lite* が使用できるリソース

リソース	デフォルト	最小/最大
CPU	1 % （注） 1 % は非標準。800 CPU ユニット/システム CPU ユニットの全体	1/100 %
メモリ	256 MB	256/256 MB

デバイスでのゲスト シェルへのアクセス

ネットワーク管理者は、IOS コマンドを使用して、ゲスト シェル内のファイルおよびユーティリティを管理することができます。

ゲスト シェルのインストール中に、SSH アクセスがキー ベースの認証でセットアップされます。ゲスト シェルへのアクセスは、IOS の最も高い特権（15）を持つユーザに制限されます。このユーザは、sudo の実行者である guestshell Linux ユーザとして Linux コンテナへのアクセスを許可され、すべてのルート操作を実行できます。ゲスト シェルから実行されるコマンドは、ユーザが IOS 端末にログインしたときと同じ特権で実行されます。

ゲスト シェル プロンプトでは、標準的な Linux コマンドを実行できます。

管理ポートを介してのゲスト シェルへのアクセス

ゲスト シェルは、デフォルトで、アプリケーションによる管理ネットワークへのアクセスを許可します。ユーザは、ゲスト シェル内から管理 VRF のネットワーキング設定を変更することはできません。

（注） 管理ポートがないプラットフォームの場合、VirtualPortGroup を IOS 設定内のゲスト シェルに関連付けることができます。詳細については、「VirtualPortGroup の設定例」の項を参照してください。

ゲスト シェルでのスタッキング

ゲスト シェルがインストールされている場合、フラッシュのファイルシステムには、gs_script ディレクトリが自動的に作成されます。このディレクトリは、スタック メンバー間で同期されます。切り替え時には、gs_script ディレクトリの内容のみが、すべてのスタック メンバー間で同期されます。ハイ アベイラビリティでの切り替えの際にデータを保持するには、このディレクトリにデータを格納します。

ハイ アベイラビリティでの切り替えの際には、新しいアクティブ デバイスは、それぞれのゲスト シェル インストールを作成します。古いファイル システムは維持されません。ゲスト シェルの状態は、切り替え時に維持されます。

IOx の概要

IOx は Cisco が開発したエンド ツー エンド アプリケーション フレームワークであり、Cisco ネットワーク プラットフォーム上のさまざまなタイプのアプリケーションに対し、アプリケーション ホスティング機能を提供します。Cisco ゲスト シェルは特殊なコンテナ展開であり、システムの開発および使用に役立つアプリケーションの 1 つです。

IOxは、構築済みアプリケーションをパッケージ化し、それらをターゲットデバイス上にホストする開発者の作業を支援する一連のサービスを提供することにより、アプリケーションのライフサイクル管理とデータ交換を容易化します。IOxのライフサイクル管理には、アプリケーションおよびデータの配布、展開、ホスティング、開始、停止（管理）、およびモニタが含まれます。IOx サービスにはアプリケーションの配布および管理ツールも含まれており、ユーザがアプリケーションを発見して IOx フレームワークに展開するのに役立ちます。

アプリケーション ホスティングは、次の機能を提供します。

- ネットワークの不均質性の遮蔽。
- デバイス上にホストされているアプリケーションのライフサイクルをリモートで管理する IOx アプリケーション プログラミング インターフェイス（API）。
- 一元的なアプリケーション ライフ サイクル管理。
- クラウド ベースの開発。

例：ゲスト シェルのネットワーキング設定

ゲスト シェルのネットワーキングでは、次の設定が必要です。

- ドメイン ネーム システム（DNS）の設定
- プロキシの設定
- プロキシの設定を使用するための YUM または PIP の設定

ゲスト シェルを有効にする方法

IOx の管理

始める前に

IOxは開始まで最長で2分かかります。ゲスト シェルを正常に有効にするには、CAF、IOXman、およびLibirtd サービスが実行している必要があります。

手順

	コマンドまたはアクション	目的
ステップ1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ2	**configure terminal** 例： `Device# configure terminal`	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3	**iox** 例： `Device(config)# iox`	IOx サービスを設定します。
ステップ4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ5	**show iox-service** 例： `Device# show iox-service`	IOx サービスのステータスを表示します。
ステップ6	**show app-hosting list** 例： `Device# show app-hosting list`	デバイスに対して有効になっている app-hosting サービスのリストを表示します。

次のタスク

次に、ISR 4000 シリーズ ルータでの **show iox-service** コマンドの出力例を示します。

```
Device# show iox-service

Virtual Service Global State and Virtualization Limits:
```

```
Infrastructure version : 1.7
Total virtual services installed : 0
Total virtual services activated : 0

Machine types supported   : KVM, LXC
Machine types disabled    : none

Maximum VCPUs per virtual service : 6
Resource virtualization limits:
Name                         Quota     Committed     Available
------------------------------------------------------------------
system CPU (%)                  75             0            75
memory (MB)                  10240             0         10240
bootflash (MB)                1000             0          1000
harddisk (MB)                20000             0         18109
volume-group (MB)           190768             0        170288


IOx Infrastructure Summary:
---------------------------
IOx service (CAF)    : Running
IOx service (HA)     : Not Running
IOx service (IOxman) : Running
Libvirtd             : Running
```

次に示すのは、Catalyst 3850 シリーズ スイッチでの **show iox-service** コマンドの短縮された出力例です。

```
Device# show iox-service

IOx Infrastructure Summary:
---------------------------
IOx service (CAF)    : Running
IOx service (HA)     : Running
IOx service (IOxman) : Running
Libvirtd             : Running
```

次に、**show app-hosting list** コマンドの出力例を示します。

```
Device# show app-hosting list

App id                           State
---------------------------------------------------------
guestshell                       RUNNING
```

ゲスト シェルの管理

（注） VirtualPortGroups はルーティング プラットフォームでのみサポートされています。

始める前に

ゲスト シェルのアクセスが機能するには、IOx が構成されて実行している必要があります。IOx が構成されていない場合は、IOx の構成を求めるメッセージが表示されます。IOx を削除すると、ゲスト シェルにもアクセスできなくなります。ただし rootfs は影響を受けません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	• **guestshell enable** 例： `Device# guestshell enable`	ゲスト シェル サービスの有効化。 （注） • **guestshell enable** コマンドは、ネットワーキングに管理 Virtual Routing and Forwarding（VRF）インスタンスを使用します。 • フロント パネル ネットワー キングに VirtualPortGroups（VPG）を使用している場合は、まず VPG を構成する必要があります。 • ゲスト IP アドレスとゲートウェイ IP アドレスは同じサブネット内にある必要があります。
ステップ 3	**guestshell run** *linux-executable* 例： `Device# guestshell run python`	ゲスト シェルで Linux プログラムを実行します。
ステップ 4	**guestshell run bash** 例： `Device# guestshell run bash`	Bash シェルを開始して、ゲスト シェルにアクセスします。
ステップ 5	**guestshell disable** 例： `Device# guestshell disable`	ゲスト シェル サービスを無効化します。

	コマンドまたはアクション	目的
ステップ 6	**guestshell destroy** 例： `Device# guestshell destroy`	ゲスト シェル サービスを非アクティブ化して、アンインストールします。

ゲスト シェルの有効化と実行

guestshell enable コマンドは、ゲスト シェルをインストールします。このコマンドは、無効化されているゲスト シェルを再アクティブ化する際にも使用されます。

ゲスト シェルが有効化された状態でシステムをリロードすると、ゲスト シェルは有効化されたままになります。

（注） **guestshell enable** コマンドを使用する前に、IOx を設定しておく必要があります。

guestshell run bash コマンドは、ゲスト シェルの bash プロンプトを開きます。このコマンドを動作させるには、ゲスト シェルが事前に有効化されていることが必要です。

（注） 次のメッセージがコンソールに表示される場合、IOx が有効化されていません。「**show iox-service** コマンドの出力をチェックして、IOx の状態を確認してください」

```
The process for the command is not responding or is otherwise unavailable
```

ゲスト シェルの無効化と破棄

guestshell disable コマンドを使用することで、ゲスト シェルを終了して無効化できます。ゲスト シェルが無効化された状態でシステムをリロードすると、ゲスト シェルは無効化されたままになります。

guestshell destroy コマンドは、フラッシュのファイル システムから rootfs を削除します。すべてのファイル、データ、インストールされている Linux アプリケーション、およびカスタムの Python ツールとユーティリティが削除され、回復できなくなります。

Python インタープリタのアクセス

Python はインタラクティブに使用できますが、Python スクリプトをゲスト シェルで実行することもできます。**guestshell run python** コマンドを使用してゲスト シェルで Python インタープリタを起動し、Python 端末を開きます。

（注） **guestshell run** コマンドは、Linux 実行可能ファイルの実行に相当する IOS であり、IOS からの Python スクリプトの実行時に絶対パスを指定します。次の例は、コマンドの絶対パスを指定する方法を示しています。

```
Guestshell run python /flash/sample_script.py parameter1 parameter2
```

ゲスト シェルの設定例

例：ゲスト シェルの管理

次の例では、Catalyst 3850 シリーズ スイッチ上でゲスト シェルを有効にする方法を示しています。

```
Device> enable
Device# guestshell enable

Management Interface will be selected if configured
Please wait for completion
Guestshell enabled successfully

Device# guestshell run python

Python 2.7.11 (default, Feb 21 2017, 03:39:40)
[GCC 5.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.

Device# guestshell run bash

[guestshell@guestshell ~]$

Device# guestshell disable

Guestshell disabled successfully

Device# guestshell destroy

Guestshell destroyed successfully
```

VirtualPortGroup 設定の例

ゲスト シェル ネットワーキングに VirtualPortGroup インターフェイスを使用する場合、VirtualPortGroup インターフェイスには設定済みの静的 IP アドレスが必要です。フロント ポート インターフェイスはインターネットに接続されている必要があり、ネット

ワーク アドレス変換（NAT）は VirtualPortGroup とフロント パネル ポートの間で設定されている必要があります。

次に示すのは、VirtualPortGroup の設定例です。

```
Device> enable
Device# configure terminal
Device(config)# interface VirtualPortGroup 0
Device(config-if)# ip address 192.168.35.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/0/3
Device(config-if)# ip address 10.0.12.19 255.255.0.0
Device(config-if)# ip nat outside
Device(config-if)# negotiation auto
Device(config-if)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1
Device(config)# ip route 10.0.0.0 255.0.0.0 10.0.0.1
!Port forwarding to use ports for SSH and so on.
Device(config)# ip nat inside source static tcp 192.168.35.2 7023 10.0.12.19 7023
extendable
Device(config)# ip nat outside source list NAT_ACL interface GigabitEthernet 0/0/3
overload
Device(config)# ip access-list standard NAT_ACL
Device(config-std-nacl)# permit 192.168.0.0 0.0.255.255
Device(config-std-nacl)# exit
Device(config)# exit
Device#
```

例：ゲスト シェルの使用

ゲスト シェル プロンプトから Linux のコマンドを実行できます。次の例は、一部の Linux コマンドの使用法を示しています。

```
[guestshell@guestshell~]$ pwd
/home/guestshell

[guestshell@guestshell~]$ whoami
guestshell

[guestshell@guestshell~]$ uname -a
Linux guestshell 3.10.101.cge-rt110 #1 SMP Sat Feb 11 00:33:02
PST 2017 mips64 GNU/Linux
```

Catalyst 3650 および Catalyst 3850 シリーズ スイッチには、BusyBox が提供する定義された一連の Linux 実行可能ファイルがあり、Cisco 4000 シリーズ サービス統合型ルータには、CentOS Linux リリース 7.1.1503 が提供するコマンドがあります。

次の例は、Catalyst 3850 シリーズ スイッチ上での **dohost** コマンドの使用を示しています。

```
[guestshell@guestshell ~]$ dohost "show version"

Cisco IOS Software [Everest], Catalyst L3 Switch Software [CAT3K_CAA-UNIVERSALK9-M),
Experimental Version 16.5.2017200014[v165_throttle-BLD-
 BLD_V165_THROTTLE_LATEST_20170531_192849 132]
```

（注） **dohost** コマンドには、**ip http server** コマンドがデバイス上で設定されていることが必要です。

例：ゲスト シェルのネットワーキング設定

ゲスト シェルのネットワーキングでは、次の設定が必要です。

- ドメイン ネーム システム（DNS）の設定
- プロキシの設定
- プロキシの設定を使用するための YUM または PIP の設定

ゲスト シェルの DNS 設定の例

ゲスト シェルのサンプル DNS 構成は次のとおりです。

```
[guestshell@guestshell ~]$ cat/etc/resolv.conf
nameserver 192.0.2.1

Other Options:
[guestshell@guestshell ~]$ cat/etc/resolv.conf
domain cisco.com
search cisco.com
nameserver 192.0.2.1
search cisco.com
nameserver 198.51.100.1
nameserver 172.16.0.6
domain cisco.com
nameserver 192.0.2.1
nameserver 172.16.0.6
nameserver 192.168.255.254
```

例：プロキシ環境変数の設定

ネットワークがプロキシの背後にある場合は、Linux でプロキシ変数を設定します。必要な場合は、環境にこれらの変数を追加します。

次の例は、プロキシ変数を設定する方法を示しています。

```
[guestshell@guestshell ~]$cat /bootflash/proxy_vars.sh
export http_proxy=http://proxy.example.com:80/
export https_proxy=http://proxy.example.com:80/
export ftp_proxy=http://proxy.example.com:80/
export no_proxy=example.com
export HTTP_PROXY=http://proxy.example.com:80/
export HTTPS_PROXY=http://proxy.example.com:80/
export FTP_PROXY=http://proxy.example.com:80/
guestshell ~] source /bootflash/proxy_vars.sh
```

例：プロキシ設定用の Yum および PIP の構成

次の例は、プロキシ環境変数の設定に Yum を使用する方法を示しています。

```
cat /etc/yum.conf | grep proxy
[guestshell@guestshell~]$ cat/bootflash/yum.conf | grep proxy
proxy=http://proxy.example.com:80/
```

PIP のインストールでは、プロキシ設定に使用される環境変数が選択されます。PIP インストールには -E オプションを指定した sudo を使用します。環境変数が設定されていない場合は、次の例に示すように PIP コマンドでそれらを明示的に定義します。

```
sudo pip --proxy http://proxy.example.com:80/install requests
sudo pip install --trusted-bost pypi.example.com --index-url
http://pypi.example.com/simple requests
```

次の例では、Python の PIP インストールを使用する方法を示します。

```
Sudo -E pip install requests
[guestshell@guestshell ~]$ python
Python 2.17.11 (default, Feb 3 2017, 19:43:44)
[GCC 4.7.0] on linux2
Type "help", "copyright", "credits" or "license" for more information
>>>import requests
```

ゲスト シェルに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
	『Programmability Command Reference, Cisco IOS XE Everest 16.6.1』
Python モジュール	『CLI Python モジュール』
ゼロ タッチ プロビジョニング	『ゼロ タッチ プロビジョニング』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

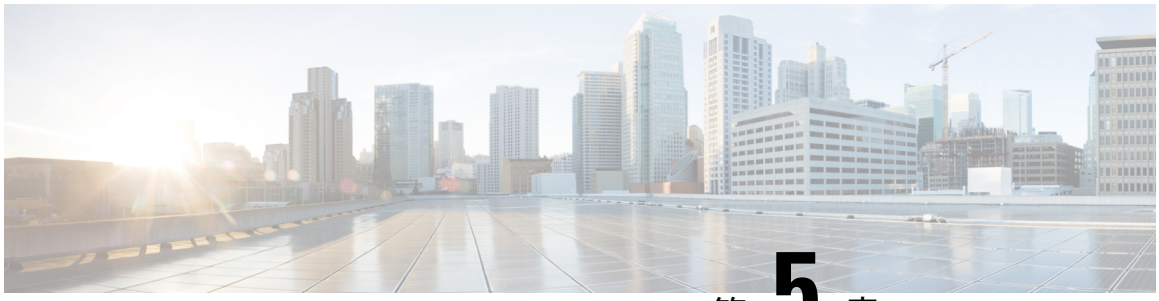
ゲスト シェルの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *9:* ゲスト シェルの機能情報

機能名	リリース	機能情報
ゲスト シェル	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	ゲスト シェルは、お客様がシスコ スイッチの自動制御および管理のためのカスタム Python アプリケーションを実行できる、埋め込み Linux 環境であるセキュア コンテナです。システムの自動化されたプロビジョニングも含まれます。このコンテナ シェルは、ホスト デバイスから分離された安全な環境を提供します。ユーザはそこで、スクリプトまたはソフトウェア パッケージをインストールし、実行することができます。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 5 章

Python API

Python プログラマビリティは、Python API をサポートしています。

• Python の使用 （63 ページ）

Python の使用

Cisco Python モジュール

シスコが提供する Python モジュールでは、EXEC および設定コマンドを実行するアクセス権が提供されます。 **help()** コマンドを入力すると、Cisco Python モジュールの詳細が表示されます。 **help()** コマンドは Cisco CLI モジュールのプロパティを表示します。

次の例は、Cisco Python モジュールに関する情報を示します。

```
Device# guestshell run python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> >>> from cli import cli,clip,configure,configurep, execute, executep
>>> help(configure)
Help on function configure in module cli:

configure(configuration)
Apply a configuration (set of Cisco IOS CLI config-mode commands) to the device
and return a list of results.

configuration = '''interface gigabitEthernet 0/0
no shutdown'''

# push it through the Cisco IOS CLI.
try:
results = cli.configure(configuration)
print "Success!"
except CLIConfigurationError as e:
print "Failed configurations:"
for failure in e.failed:
print failure
```

```
Args:
configuration (str or iterable): Configuration commands, separated by newlines.

Returns:
list(ConfigResult): A list of results, one for each line.

Raises:
CLISyntaxError: If there is a syntax error in the configuration.
```

```
>>> help(configurep)
Help on function configurep in module cli:

configurep(configuration)
Apply a configuration (set of Cisco IOS CLI config-mode commands) to the device
and prints the result.

configuration = '''interface gigabitEthernet 0/0
no shutdown'''

# push it through the Cisco IOS CLI.
configurep(configuration)

Args:
configuration (str or iterable): Configuration commands, separated by newlines.
```

```
>>> help(execute)
Help on function execute in module cli:

execute(command)
Execute Cisco IOS CLI exec-mode command and return the result.

command_output = execute("show version")

Args:
command (str): The exec-mode command to run.

Returns:
str: The output of the command.

Raises:
CLISyntaxError: If there is a syntax error in the command.
```

```
>>> help(executep)
Help on function executep in module cli:

executep(command)
Execute Cisco IOS CLI exec-mode command and print the result.

executep("show version")

Args:
command (str): The exec-mode command to run.
```

```
>>> help(cli)
Help on function cli in module cli:

cli(command)
    Execute Cisco IOS CLI command(s) and return the result.

    A single command or a delimited batch of commands may be run. The
    delimiter is a space and a semicolon, " ;". Configuration commands must be
    in fully qualified form.
```

```
    output = cli("show version")
    output = cli("show version ; show ip interface brief")
    output = cli("configure terminal ; interface gigabitEthernet 0/0 ; no shutdown")

    Args:
        command (str): The exec or config CLI command(s) to be run.

    Returns:
        string: CLI output for show commands and an empty string for
            configuration commands.

    Raises:
        errors.cli_syntax_error: if the command is not valid.
        errors.cli_exec_error: if the execution of command is not successful.


>>> help(clip)
Help on function clip in module cli:

clip(command)
    Execute Cisco IOS CLI command(s) and print the result.

    A single command or a delimited batch of commands may be run. The
    delimiter is a space and a semicolon, " ;". Configuration commands must be
    in fully qualified form.

    clip("show version")
    clip("show version ; show ip interface brief")
    clip("configure terminal ; interface gigabitEthernet 0/0 ; no shutdown")

    Args:
        command (str): The exec or config CLI command(s) to be run.
```

IOS CLI コマンドを実行するための Cisco Python モジュール

（注） Python を実行するには、ゲスト シェルが有効である必要があります。詳細については、「ゲスト シェル」の章を参照してください。

Python プログラミング言語は CLI コマンドを実行できる 6 つの関数を使用します。これらの関数は、Python CLI モジュールから利用できます。これらの関数を使用するには、**import cli** コマンドを実行します。これらの関数が機能するには、**ip http server** コマンドが有効になっている必要があります。

これらの関数の引数は CLI コマンドの文字列です。Python インタープリタ経由で CLI コマンドを実行するには、次の 6 つの関数のいずれかの引数文字列として CLI コマンドを入力します。

- **cli.cli(command)**：この関数は IOS コマンドを引数として取り、IOS パーサーからコマンドを実行し、結果のテキストを返します。このコマンドの形式が正しくない場合、Python の例外が発生します。次に、**cli.cli(command)** 関数の出力例を示します。

```
>>> import cli
>>> cli.clip('configure terminal; interface loopback 10; ip address
```

```
 10.10.10.10 255.255.255.255')
*Mar 13 18:39:48.518: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback10,
changed state to up
>>> cli.clip('show clock')
'\n*18:11:53.989 UTC Mon Mar 13 2017\n'
>>> output=cli.cli('show clock')
>>> print(output)
*18:12:04.705 UTC Mon Mar 13 2017
```

- **cli.clip(command)**：この関数は **cli.cli(command)** 関数と機能はまったく同じです。ただし結果のテキストを（返すのではなく）stdout に出力する点が異なります。次に、**cli.clip(command)** 関数の出力例を示します。

```
>>> cli
>>> cli.clip('configure terminal; interface loopback 11; ip address
 10.11.11.11 255.255.255.255')
*Mar 13 18:42:35.954: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback11,
changed state to up
*Mar 13 18:42:35.954: %LINK-3-UPDOWN: Interface Loopback11, changed state to up
>>> cli.clip('show clock')
*18:13:35.313 UTC Mon Mar 13 2017
>>> output=cli.clip('show clock')
*18:19:26.824 UTC Mon Mar 13 2017
>>> print (output)
None
```

- **cli.execute(command)**：この関数は単一の EXEC コマンドを実行して出力を返します。ただし結果のテキストは出力しません。このコマンドの一部としてセミコロンまたは改行を使用することは許可されません。この関数を複数回実行するには、for-loop が指定された Python リストを使用します。次に、 **cli.execute(command)**

  関数の出力例を示します。

```
>>> cli.execute("show clock")
'15:11:20.816 UTC Thu Jun 8 2017'
>>>
>>> cli.execute('show clock'; 'show ip interface brief')
  File "<stdin>", line 1
    cli.execute('show clock'; 'show ip interface brief')
                            ^
SyntaxError: invalid syntax
>>>
```

- **cli.executep(command)**：この関数は単一のコマンドを実行して、結果のテキストを（返すのではなく）stdout に出力します。次に、**cli.executep(command)** 関数の出力例を示します。

```
>>> cli.executep('show clock')
*18:46:28.796 UTC Mon Mar 13 2017
>>> output=cli.executep('show clock')
*18:46:36.399 UTC Mon Mar 13 2017
>>> print(output)
```

```
None
```

- **cli.configure(command)**：この関数は、コマンドで使用できる設定によりデバイスを設定します。これは次に示すように、コマンドとその結果が含まれる名前付きタプルのリストを返します。

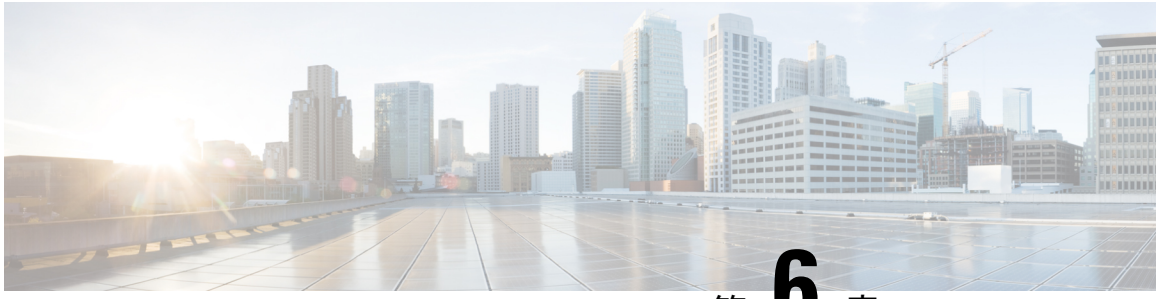
```
[Think: result = (bool(success), original_command, error_information)]
```

コマンド パラメータは複数行に入力することができ、**show running-config** コマンドの出力に表示されているのと同じ形式にすることができます。次に、**cli.configure(command)** 関数の出力例を示します。

```
>>>cli.configure(["interface GigabitEthernet1/0/7", "no shutdown",
"end"])
[ConfigResult(success=True, command='interface GigabitEthernet1/0/7',
line=1, output='', notes=None), ConfigResult(success=True, command='no shutdown',
line=2, output='', notes=None), ConfigResult(success=True, command='end',
line=3, output='', notes=None)]
```

- **cli.configurep(command)**：この関数は **cli.configure(command)** 関数と機能はまったく同じです。ただし結果のテキストを（返すのではなく）stdout に出力する点が異なります。次に、**cli.configurep(command)** 関数の出力例を示します。

```
>>> cli.configurep(["interface GigabitEthernet1/0/7", "no shutdown",
 "end"])
Line 1 SUCCESS: interface GigabitEthernet1/0/7
Line 2 SUCCESS: no shut
Line 3 SUCCESS: end
```



第 6 章

CLI Python モジュール

Python プログラマビリティでは、CLI を使用して IOS と対話できる Python モジュールを提供しています。

Python CLI モジュールについて

Python について

Cisco IOS XE デバイスは、ゲスト シェル内でインタラクティブおよび非インタラクティブ（スクリプト）の両方のモードで Python バージョン 2.7 をサポートします。Python スクリプト機能により、デバイスの CLI にプログラムを使用してアクセスして、さまざまなタスク、およびゼロ タッチ プロビジョニングまたは Embedded Event Manager（EEM）アクションを実行することができます。

Python スクリプトの概要

Python は、仮想化された Linux ベースの環境であるゲスト シェルで実行されます。詳細については、「ゲスト シェル」の章を参照してください。シスコが提供する Python モジュールは、ユーザの Python スクリプトがホスト デバイス上で IOS CLI コマンドを実行することを可能にします。

対話形式の Python プロンプト

デバイス上で **guestshell run python** コマンドを実行すると、ゲスト シェル内で、対話形式の Python プロンプトが開きます。Python の対話モードでは、Cisco Python CLI モジュールから Python 機能を実行してデバイスを設定することができます。

次の例は、対話形式の Python プロンプトを有効にする方法を示しています。

```
Device# guestshell run python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>

Device#
```

Python スクリプト

Python スクリプト名を引数として Python コマンドで使用することで、Python スクリプトを非インタラクティブ モードで実行できます。Python スクリプトは、ゲスト シェル内からアクセス可能である必要があります。ゲスト シェルから Python スクリプトにアクセスするには、ゲスト シェル内にマウントされているブートフラッシュまたはフラッシュにスクリプトを保存します。

（注） Python で **import cli** が機能するように、**ip http server** コマンドを設定する必要があります。

次の Python スクリプトの例は、さまざまな CLI 関数を使用して **show** コマンドを設定および出力します。

```
Device# more flash:sample_script.py


import sys
import cli

intf= sys.argv[1:]
intf = ''.join(intf[0])

print "\n\n *** Configuring interface %s with 'configurep' function  *** \n\n" %intf
cli.configurep(["interface loopback55","ip address 10.55.55.55 255.255.255.0","no
shut","end"])

print "\n\n *** Configuring interface %s with 'configure' function  *** \n\n"
cmd='interface %s,logging event link-status ,end' % intf
cli.configure(cmd.split(','))

print "\n\n *** Printing show cmd with 'executep' function  *** \n\n"
cli.executep('show ip interface brief')

print "\n\n *** Printing show cmd with 'execute' function  *** \n\n"
output= cli.execute('show run interface %s' %intf)
print (output)

print "\n\n *** Configuring interface %s with 'cli' function  *** \n\n"
cli.cli('config terminal; interface %s; spanning-tree portfast edge default' %intf)

print "\n\n *** Printing show cmd with 'clip' function  *** \n\n"
cli.clip('show run interface %s' %intf)


To run a Python script from the Guest Shell, execute the guestshell run python
/flash/script.py command
at the device prompt.
```

The following example shows how to run a Python script from the Guest Shell:

次の例は、ゲスト シェルから Python スクリプトを実行する方法を示しています。

```
Device# guestshell run python /flash/sample_script.py loop55

 *** Configuring interface loop55 with 'configurep' function  ***

Line 1 SUCCESS: interface loopback55
Line 2 SUCCESS: ip address 10.55.55.55 255.255.255.0
Line 3 SUCCESS: no shut
Line 4 SUCCESS: end

 *** Configuring interface %s with 'configure' function  ***


 *** Printing show cmd with 'executep' function  ***

Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0     192.0.2.1       YES NVRAM  up                    up
GigabitEthernet1/0/1   unassigned      YES unset  down                  down
GigabitEthernet1/0/2   unassigned      YES unset  down                  down
GigabitEthernet1/0/3   unassigned      YES unset  down                  down
    :
          :
          :
Te1/1/4                unassigned      YES unset  down                  down
Loopback55             10.55.55.55     YES TFTP   up                    up
Loopback66             unassigned      YES manual up                    up


 *** Printing show cmd with 'execute' function  ***

Building configuration...
Current configuration : 93 bytes
!
interface Loopback55
 ip address 10.55.55.55 255.255.255.0
 logging event link-status
end

 *** Configuring interface %s with 'cli' function  ***

 *** Printing show cmd with 'clip' function  ***

Building configuration...
Current configuration : 93 bytes
!
interface Loopback55
 ip address 10.55.55.55 255.255.255.0
 logging event link-status
end
```

サポートされる Python のバージョン

ゲスト シェルは、Python バージョン 2.7 をプリインストールしています。ゲスト シェルは、仮想化された Linux ベースの環境であり、Cisco デバイスの自動制御と管理のための Python ア

プリケーションを含む、カスタム Linux アプリケーションを実行するように設計されています。Montavista CGE7 がインストールされたプラットフォームは Python バージョン 2.7.11 をサポートし、CentOS 7 がインストールされたプラットフォームは Python バージョン 2.7.5 をサポートします。

次の表は、Python の各バージョンおよびサポート対象のプラットフォームに関する情報を示しています。

表 *10 : Python* バージョン サポート

Python のバージョン	**プラットフォーム**
Python バージョン 2.7.5	Cisco Catalyst 3650 シリーズ スイッチおよび Cisco Catalyst 3850 シリーズ スイッチを除くすべてのサポート対象プラットフォーム。
Python バージョン 2.7.11	• Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ
Python バージョン 3.6	Cisco IOS XE Amsterdam 17.1.1 以降のリリースでサポートされています。 Cisco IOS XE Amsterdam 17.1.1 および Cisco IOS XE Amsterdam 17.2.1 では、Python V2 がデフォルトです。ただし、Cisco IOS XE Amsterdam 17.3.1 以降のリリースでは、Python V3 がデフォルトです。 （注） Cisco Catalyst 9200 シリーズ スイッチは、Cisco IOS XE Amsterdam 17.1.1 および Cisco IOS XE Amsterdam 17.2.1 で Python Version 3.6 をサポートしていません。Cisco Catalyst 9200 シリーズ スイッチは、Cisco IOS XE Amsterdam 17.3.1 以降のリリースで Python V3 をサポートしています。 （注） Cisco Catalyst 3650 シリーズ スイッチおよび Cisco Catalyst 3850 シリーズスイッチではサポートされていません。

CentOS 7 がインストールされたプラットフォームは、オープン ソース リポジトリからの Redhat Package Manager（RPM）のインストールをサポートします。

Cisco CLI Python モジュールの更新

Cisco CLI Python モジュールおよび EEM モジュールは、デバイスにインストール済みです。ただし、Yum または事前にパッケージ化されているバイナリのいずれかを使用して Python のバージョンを更新する場合は、シスコが提供する CLI モジュールも更新する必要があります。

（注） Python バージョン 2 がすでにあるデバイスで Python バージョン 3 への更新を行うと、デバイス上には両方のバージョンの Python が存在するようになります。Python を実行するには、次の IOS コマンドのいずれかを使用します。

- **guestshell run python2** コマンドは、Python バージョン 2 を有効化します。
- **guestshell run python3** コマンドは、Python バージョン 3 を有効化します。
- **guestshell run python** コマンドは、Python バージョン 2 を有効化します。

Python のバージョンを更新するには、次の方法のいずれかを使用します。

- スタンドアロン tarball のインストール
- CLI モジュールのための PIP のインストール

CLI Python モジュールに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
ゲスト シェル	ゲスト シェル
EEM Python モジュール	EEM の Python スクリプト

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

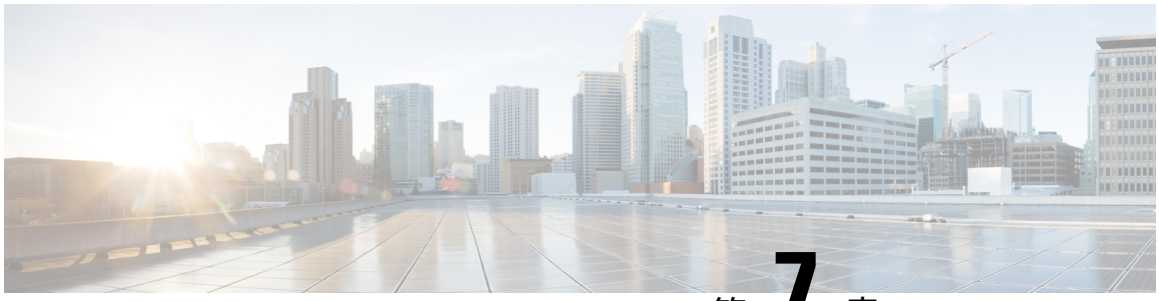
CLI Python モジュールの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *11 : CLI Python* モジュールの機能情報

機能名	リリース	機能情報
CLI Python モジュール	Cisco IOS XE Everest 16.5.1a	Python プログラマビリティでは、ユーザが CLI を使用して IOS と対話できるようにする Python モジュールが提供されます。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 7 章

EEM Python モジュール

組み込みイベント マネージャ（EEM）ポリシーは、Python スクリプトをサポートします。Python スクリプトは、EEM アプレットで EEM アクションの一部として実行できます。

EEM Python モジュールの前提条件

ゲスト シェルは、コンテナ内で機能する必要があります。ゲスト シェルは、デフォルトでは有効になっていません。詳細については、ゲスト シェル機能の説明を参照してください。

EEM Python モジュールについて

EEM の Python スクリプト

組み込みイベント マネージャ（EEM）ポリシーは、Python スクリプトをサポートします。Python スクリプトを EEM ポリシーとして登録し、対応するイベントが発生したときに、登録済みの Python スクリプトを実行することができます。EEM Python スクリプトには、EEM TCL ポリシーと同じイベント仕様の構文があります。

設定済みの EEM ポリシーは、ゲスト シェル内で実行します。ゲスト シェルは、仮想化された Linux ベースの環境であり、Cisco デバイスの自動制御と管理のための Python アプリケーションを含む、カスタム Linux アプリケーションを実行するように設計されています。ゲスト シェル コンテナは、Python インタープリタを提供します。

EEM Python パッケージ

EEM Python パッケージを Python スクリプトにインポートすると、EEM に固有の拡張機能を実行できます。

（注） EEM Python パッケージは、EEM Python スクリプト内でのみ使用できます（パッケージは EEM に登録でき、スクリプトの最初の行に EEM イベント仕様が記載されます）。標準的な Python スクリプト（Python スクリプト名を使用して実行される）では使用できません。

Python パッケージには、次のアプリケーションプログラミングインターフェイス（API）が含まれています。

- アクション API：EEM アクションを実行するもので、デフォルトのパラメータがあります。
- CLI 実行 API：IOS コマンドを実行し、出力を返します。次に、CLI 実行 API のリストを示します。
  - eem_cli_open()
  - eem_cli_exec()
  - eem_cli_read()
  - eem_cli_read_line()
  - eem_cli_run()
  - eem_cli_run_interactive()
  - eem_cli_read_pattern()
  - eem_cli_write()
  - eem_cli_close()
- 環境変数にアクセスする API：組み込みまたはユーザ定義の変数のリストを取得します。次に、環境変数にアクセスする API を示します。
  - eem_event_reqinfo ()：組み込み変数のリストを返します。
  - eem_user_variables()：引数の現在の値を返します。

Python がサポートする EEM アクション

Python パッケージ（EEM スクリプト内でのみ使用可能で、標準的な Python スクリプトでは使用不可）では、次の EEM アクションをサポートしています。

- Syslog メッセージの印刷
- SNMP トラップの送信

- ボックスのリロード
- スタンバイ デバイスへの切り替え
- ポリシーの実行
- トラック オブジェクトの読み取り
- トラック オブジェクト セット
- Cisco ネットワーキング サービスのイベントの生成

EEM Python パッケージは、EEM アクションを実行するため、インターフェイスを公開します。これらのアクションは Python スクリプトを使用して呼び出すことができ、Cisco Plug N Play（PnP）経由で Python パッケージからアクション ハンドラに転送されます。

EEM 変数

EEM ポリシーは、次の種類の変数を持つことができます。

- イベント固有の組み込み変数：ポリシーをトリガーしたイベントの詳細が設定される事前定義の変数のセット。eem_event_reqinfo () API は、組み込み変数のリストを返します。これらの変数は、ローカルマシンに保存してローカル変数として使用することができます。ローカル変数に対する変更は、組み込み変数に反映されません。
- ユーザ定義の変数：定義およびポリシーでの使用が可能な変数。これらの変数の値は、Python スクリプト内で参照できます。スクリプトを実行する際に、変数の最新の値が使用可能であることを確認してください。eem_user_variables() API は、API で入力された引数の現在の値を返します。

EEM CLI ライブラリのコマンド拡張

EEM 内では、Python スクリプトを動作させるため、次の CLI ライブラリ コマンドを使用できます。

- eem_cli_close()：EXEC プロセスをクローズし、コマンドに接続された、VTY および指定されたチャネル ハンドラをリリースします。
- eem_cli_exec：指定されたチャネル ハンドラにコマンドを記述し、コマンドを実行します。次に、チャネルからコマンドの出力を読み取り、出力を返します。
- eem_cli_open：VTY を割り当て、EXEC CLI セッションを作成し、VTY をチャネル ハンドラに接続します。チャネル ハンドラを含む配列を返します。
- eem_cli_read()：読み取られている内容でデバイス プロンプトのパターンが発生するまで、指定された CLI のチャネル ハンドラからコマンド出力を読み取ります。一致するまで、読み取られたすべての内容を返します。
- eem_cli_read_line()：指定された CLI のチャネル ハンドラから、コマンド出力の 1 行を読み取ります。読み取られた行を返します。

- eem_cli_read_pattern()：読み取られている内容でパターンが発生するまで、指定された CLI のチャネル ハンドラからコマンド出力を読み取ります。一致するまで、読み取られたすべての内容を返します。
- eem_cli_run()：clist にある項目を繰り返し、それぞれが、イネーブル モードで実行されるコマンドであることを前提とします。正常に実行されると、実行されたすべてのコマンドの出力を返します。失敗すると、エラーを返します。
- eem_cli_run_interactive()：3 つの項目がある clist のサブリストを用意します。正常に実行されると、実行されたすべてのコマンドの出力を返します。失敗すると、エラーを返します。可能な場合には、配列も使用します。予測と応答を別々に保持することによって、より簡単に後で読み取ることができます。
- eem_cli_write()：指定された CLI チャネル ハンドラに対して実行されるコマンドを書き込みます。CLI チャネル ハンドラによって、コマンドが実行されます。

EEM Python ポリシーの設定方法

Python スクリプトが動作できるようにするには、ゲスト シェルを有効化する必要があります。詳細については、「ゲスト シェル」の章を参照してください。

Python ポリシーの登録

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**event manager directory user policy** *path* 例： `Device(config)# event manager directory user policy flash:/user_library`	ユーザ ライブラリ ファイルまたはユーザ定義 EEM ポリシーの保存に使用するディレクトリを指定します。 （注） 指定されたパスにポリシーが必要です。たとえば、この手順では、eem_script.py ポリシーが flash:/user_library フォルダーまたはパスで使用できます。

	コマンドまたはアクション	目的
ステップ 4	**event manager policy** *policy-filename* 例： `Device(config)# event manager policy eem_script.py`	EEM ポリシーを EEM に登録します。 • ポリシーは、ファイル拡張子に基づいて解析されます。ファイル拡張子は .py で、ポリシーは Python ポリシーとして登録されます。 • EEM は、ポリシーそのものに含まれるイベント仕様に基づいてポリシーをスケジューリングし、実行します。**event manager policy** コマンドが呼び出されると、EEM はポリシーを確認し、指定されたイベントが発生した場合に実行されるように登録します。
ステップ 5	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show event manager policy registered** 例： `Device# show event manager policy registered`	保留 EEM ポリシーを表示します。
ステップ 7	**show event manager history events** 例： `Device# show event manager history events`	トリガーされた EEM イベントを表示します。

例

次に、**show event manager policy registered** コマンドの出力例を示します。

```
Device# show event manager policy registered

No.  Class     Type    Event Type          Trap  Time Registered           Name
1    script    user    multiple            Off   Tue Aug 2 22:12:15 2016   multi_1.py
 1: syslog: pattern {COUNTER}
 2: none: policyname {multi_1.py} sync {yes}
 trigger delay 10.000
  correlate event 1 or event 2
  attribute tag 1 occurs 1
 nice 0 queue-priority normal maxrun 100.000 scheduler rp_primary Secu none


2    script    user    multiple            Off   Tue Aug 2 22:12:20 2016   multi_2.py
 1: syslog: pattern {COUNTER}
 2: none: policyname {multi_2.py} sync {yes}
```

```
 trigger
  correlate event 1 or event 2
 nice 0 queue-priority normal maxrun 100.000 scheduler rp_primary Secu none


3    script    user    multiple            Off   Tue Aug 2 22:13:31 2016   multi.tcl
 1: syslog: pattern {COUNTER}
 2: none: policyname {multi.tcl} sync {yes}
 trigger
  correlate event 1 or event 2
  attribute tag 1 occurs 1
 nice 0 queue-priority normal maxrun 100.000 scheduler rp_primary Secu none
```

EEM アプレット アクションの一部としての Python スクリプトの実行

Python スクリプト：eem_script.py

アクション コマンドを使用することで、EEM アプレットに Python スクリプトを含めることができます。この例では、ユーザは標準 Python スクリプトを EEM アクションの一部として実行しようとしています。ただし、EEM Python パッケージは標準 Python スクリプトでは使用できません。IOS の標準 Python スクリプトには from cli import cli,clip という名前のパッケージがあり、そのパッケージは IOS コマンドを実行するために使用できます。

```
import sys
from cli import cli,clip,execute,executep,configure,configurep

intf= sys.argv[1:]
intf = ''.join(intf[0])


print ('This script is going to unshut interface %s and then print show ip interface
brief'%intf)

if intf == 'loopback55':
configurep(["interface loopback55","no shutdown","end"])
else :
cmd='int %s,no shut ,end' % intf
configurep(cmd.split(','))

executep('show ip interface brief')
```

次に、**guestshell run python** コマンドの出力例を示します。

```
Device#  guestshell run python /flash/eem_script.py loop55

This script is going to unshut interface loop55 and then print show ip interface brief
Line 1 SUCCESS: int loop55
Line 2 SUCCESS: no shut
Line 3 SUCCESS: end
Interface IP-Address OK? Method Status Protocol
Vlan1 unassigned YES NVRAM administratively down down
GigabitEthernet0/0 5.30.15.37 YES NVRAM up up
GigabitEthernet1/0/1 unassigned YES unset down down
GigabitEthernet1/0/2 unassigned YES unset down down
```

```
GigabitEthernet1/0/3 unassigned YES unset down down
GigabitEthernet1/0/4 unassigned YES unset up up
GigabitEthernet1/0/5 unassigned YES unset down down
GigabitEthernet1/0/6 unassigned YES unset down down
GigabitEthernet1/0/7 unassigned YES unset down down
GigabitEthernet1/0/8 unassigned YES unset down down
GigabitEthernet1/0/9 unassigned YES unset down down
GigabitEthernet1/0/10 unassigned YES unset down down
GigabitEthernet1/0/11 unassigned YES unset down down
GigabitEthernet1/0/12 unassigned YES unset down down
GigabitEthernet1/0/13 unassigned YES unset down down
GigabitEthernet1/0/14 unassigned YES unset down down
GigabitEthernet1/0/15 unassigned YES unset down down
GigabitEthernet1/0/16 unassigned YES unset down down
GigabitEthernet1/0/17 unassigned YES unset down down
GigabitEthernet1/0/18 unassigned YES unset down down
GigabitEthernet1/0/19 unassigned YES unset down down
GigabitEthernet1/0/20 unassigned YES unset down down
GigabitEthernet1/0/21 unassigned YES unset down down
GigabitEthernet1/0/22 unassigned YES unset down down
GigabitEthernet1/0/23 unassigned YES unset up up
GigabitEthernet1/0/24 unassigned YES unset down down
GigabitEthernet1/1/1 unassigned YES unset down down
GigabitEthernet1/1/2 unassigned YES unset down down
GigabitEthernet1/1/3 unassigned YES unset down down
GigabitEthernet1/1/4 unassigned YES unset down down
Te1/1/1 unassigned YES unset down down
Te1/1/2 unassigned YES unset down down
Te1/1/3 unassigned YES unset down down
Te1/1/4 unassigned YES unset down down
Loopback55 10.55.55.55 YES manual up up

Device#
Jun 7 12:51:20.549: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback55,
changed state to up
Jun 7 12:51:20.549: %LINK-3-UPDOWN: Interface Loopback55, changed state to up
```

次に示すのは、syslog へのメッセージ出力のサンプル スクリプトです。このスクリプトは、ファイルに保存され、デバイス上のファイル システムにコピーされ、イベント マネージャのポリシー ファイルを使用して登録される必要があります。

```
::cisco::eem::event_register_syslog tag "1"    pattern     COUNTER    maxrun 200

import eem
import time

eem.action_syslog("SAMPLE SYSLOG MESSAGE","6","TEST")
```

次に示すのは、EEM 環境変数を出力するサンプル スクリプトです。このスクリプトは、ファイルに保存され、デバイス上のファイル システムにコピーされ、イベント マネージャのポリシー ファイルを使用して登録される必要があります。

```
::cisco::eem::event_register_syslog tag "1"    pattern     COUNTER    maxrun 200

import eem
import time

c = eem.env_reqinfo()
```

```
print "EEM Environment Variables"
for k,v in c.iteritems():
      print "KEY : " + k + str(" ---> ") + v

print "Built in Variables"
for i,j in a.iteritems() :
      print "KEY : " + i + str(" ---> ") + j
```

EEM アプレットでの Python スクリプトの追加

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**event manager applet** *applet-name* 例： `Device(config)# event manager applet interface_Shutdown`	Embedded Event Manager（EEM）にアプレットを登録し、アプレット 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	**event** [**tag** *event-tag*] **syslog pattern** *regular-expression* 例： `Device(config-applet)# event syslog pattern "Interface Loopback55, changed state to administratively down"`	syslog メッセージのパターン一致を実行する正規表現を指定します。
ステップ 5	**action** *label* **cli command** *cli-string* 例： `Device(config-applet)# action 0.0 cli command "en"`	EEM アプレットがトリガーされたときに実行される IOS コマンドを指定します。
ステップ 6	**action** *label* **cli command** *cli-string* [**pattern** *pattern-string*] 例： `Device(config-applet)# action 1.0 cli command "guestshell run python3 /bootflash/eem_script.py loop55"`	**pattern** キーワードで指定されるアクションを指定します。 • 次の要請プロンプトに一致する正規表現パターン文字列を指定します。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-applet)# end`	アプレット コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 8	**show event manager policy active** 例： `Device# show event manager policy active`	実行している EEM ポリシーを表示します。
ステップ 9	**show event manager history events** 例： `Device# show event manager history events`	トリガーされた EEM イベントを表示します。

次のタスク

次の例では、タスクに設定されている Python スクリプトをトリガーする方法を示しています。

```
Device(config)# interface loopback 55
Device(config-if)# shutdown
Device(config-if)# end
Device#

Mar 13 10:53:22.358 EDT: %SYS-5-CONFIG_I: Configured from console by console
Mar 13 10:53:24.156 EDT: %LINK-5-CHANGED:  Line protocol on Interface Loopback55, changed
 state to down
Mar 13 10:53:27.319 EDT: %LINK-3-UPDOWN:  Interface Loopback55, changed state to
administratively down
Enter configuration commands, one per line.  End with CNTL/Z.
Mar 13 10:53:35.38 EDT:  %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback55,
changed state to up
*Mar 13 10:53:35.39 EDT %LINK-3-UPDOWN: Interface Loopback55, changed state to up
+++ 10:54:33 edi37(default) exec +++
show ip interface br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  down                  down
GigabitEthernet0/0/1   unassigned      YES unset  down                  down
GigabitEthernet0/0/2   10.1.1.31       YES DHCP   up                    up
GigabitEthernet0/0/3   unassigned      YES unset  down                  down
GigabitEthernet0       192.0.2.1       YES manual up                    up
Loopback55             198.51.100.1    YES manual up                    up
Loopback66             172.16.0.1      YES manual up                    up
Loopback77             192.168.0.1     YES manual up                    up
Loopback88             203.0.113.1     YES manual up                    up
```

EEM Python モジュールに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
EEM 設定	『*Embedded Event Manager Configuration Guide*』
EEM コマンド	『*Embedded Event Manager Command Reference*』
ゲスト シェル設定	『ゲスト シェル』

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

EEM Python モジュールの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

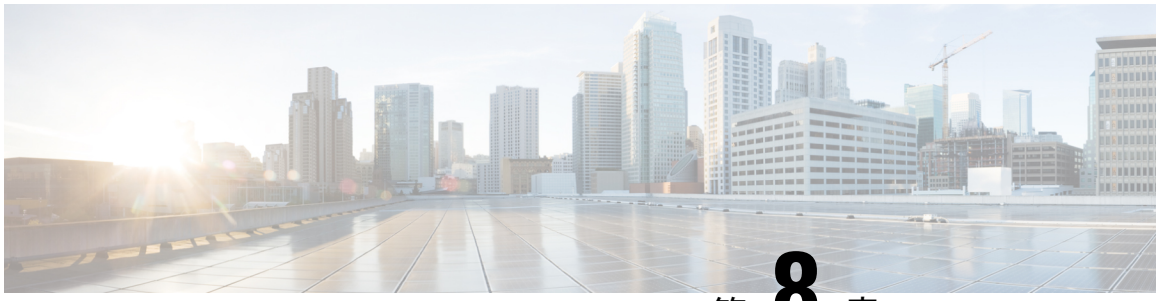
プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *12 : EEM Python* モジュールの機能情報

機能名	リリース	機能情報
EEM Python モジュール	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	この機能は、EEM ポリシーとして Python スクリプトをサポートします。 追加された新規コマンドはありません。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco ISR 4000 シリーズ サービス統合型ルータ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。

第 III 部

モデル駆動型プログラマビリティ



第 8 章

NETCONF プロトコル

NETCONF プロトコルの制約事項

NETCONF 機能は、デュアル IOSd 設定またはソフトウェア冗長性を実行中のデバイスではサポートされていません。

NETCONF プロトコルの概要

データモデルの概要：プログラムによる設定と各種の標準規格に準拠した設定

ネットワーク デバイスを管理する従来の方法は、階層的データ（設定コマンド）および運用データ（show コマンド）用のコマンドライン インターフェイス（CLI）を使用することです。ネットワーク管理の場合、特にさまざまなネットワーク デバイス間で管理情報を交換するために、Simple Network Management Protocol（SNMP）が広く使用されています。頻繁に使用されている CLI と SNMP ですが、これにはいくつかの制約事項があります。CLI は非常に独自的であり、テキストベースの仕様を理解し、解釈するには人間の介入が必要です。SNMP は、階層的データと運用データを区別しません。

これを解決するには、手作業で設定作業を行うのではなく、プログラムを使用したり、各種の標準規格に準拠してネットワーク デバイスの設定を記述します。Cisco IOS XE で動作するネットワーク デバイスは、データ モデルを使用するネットワーク上の複数のデバイスの設定の自

動化をサポートしています。データ モデルは、業界で定義された標準的な言語で開発され、ネットワークの設定とステータス情報を定義できます。

Cisco IOS XE は、Yet Another Next Generation（YANG）データ モデリング言語をサポートしています。YANG をネットワーク設定プロトコル（NETCONF）で使用すると、自動化されたプログラミング可能なネットワーク操作の望ましいソリューションが実現します。NETCONF（RFC 6241）は、クライアント アプリケーションがデバイスからの情報を要求してデバイスに設定変更を加えるために使用する XML ベースのプロトコルです。YANG は主に、NETCONF 操作で使用される設定とステート データをモデル化するために使用されます。

Cisco IOS XE では、モデル ベースのインターフェイスは、既存のデバイス CLI、Syslog、および SNMP インターフェイスと相互運用します。必要に応じて、これらのインターフェイスは、ネットワーク デバイスからノースバウンドに公開されます。YANG は、RFC 6020 に基づいて各プロトコルをモデル化するために使用されます。

（注） 開発者に分かりやすい方法で Cisco YANG モデルにアクセスするには、GitHub リポジトリを複製し、vendor/cisco サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

NETCONF

NETCONF は、ネットワーク デバイスの設定をインストール、操作、削除するためのメカニズムです。

コンフィギュレーション データとプロトコル メッセージに Extensible Markup Language（XML）ベースのデータ符号化を使用します。

NETCONF はシンプルなリモート プロシージャ コール（RPC）ベースのメカニズムを使用してクライアントとサーバ間の通信を促進します。クライアントはネットワーク マネージャの一部として実行されているスクリプトやアプリケーションです。通常、サーバはネットワーク デバイス（スイッチまたはルータ）です。サーバは、ネットワーク デバイス全体のトランスポート層としてセキュア シェル（SSH）を使用します。SSH ポート番号 830 をデフォルトのポートとして使用します。ポート番号は、設定可能なオプションです。

NETCONF は、機能の検出およびモデルのダウンロードもサポートしています。サポート対象のモデルは、ietf-netconf-monitoring モデルを使用して検出されます。各モデルに対する改定日付は、機能の応答に示されています。データ モデルは、get-schema RPC を使用して、デバイスからオプションのダウンロードとして入手できます。これらの YANG モデルを使用して、データ モデルを理解したりエクスポートしたりできます。NETCONF の詳細については、RFC 6241 を参照してください。

Cisco IOS XE Fuji 16.8.1 よりも前のリリースでは、運用データ マネージャ（ポーリングに基づく）が個別に有効になっていました。Cisco IOS XE Fuji 16.8.1 以降のリリースでは、運用データは、NETCONF を実行しているプラットフォームで動作し（設定データの仕組みと同様）、デフォルトで有効になっています。運用データのクエリまたはストリーミングに対応するコンポーネントの詳細については、GitHub リポジトリで命名規則の *-oper を参照してください。

NETCONF RESTCONF IPv6 のサポート

データ モデル インターフェイス（DMI）は IPv6 プロトコルの使用をサポートしています。DMI による IPv6 のサポートは、クライアント アプリケーションが、IPv6 アドレスを使用するサービスと通信する場合に役に立ちます。外部向けインターフェイスは、IPv4 と IPv6 の両方についてデュアルスタックをサポートします。

DMI は、ネットワーク要素の管理を容易にする一連のサービスです。NETCONF や RESTCONF などのアプリケーション層プロトコルは、ネットワークを介してこれらの DMI にアクセスします。

IPv6 アドレスが設定されていない場合でも、外部向けアプリケーションは IPv6 ソケットをリッスンし続けますが、これらのソケットは到達不能になります。

NETCONF グローバル セッションのロック

NETCONF プロトコルは、デバイス設定を管理し、デバイスの状態情報を取得するための一連の操作を提供します。NETCONF はグローバル ロックをサポートしており、NETCONF では応答しなくなったセッションを kill する機能が導入されています。

複数の同時セッションの全体にわたって一貫性を確保し、設定の競合を防ぐために、セッションのオーナーは NETCONF セッションをロックできます。NETCONF lock RPC は、コンフィギュレーション パーサーと実行コンフィギュレーション データベースをロックします。その他のすべての NETCONF セッション（ロックを所有していない）は、編集操作を実行できません。ただし、読み取り操作は実行できます。これらのロックは存続時間が短いことを意図しており、オーナーは、他の NETCONF クライアント、NETCONF 以外のクライアント（SNMP、CLI スクリプトなど）、および人間のユーザとやり取りをせずに変更を加えることができます。

アクティブセッションによって保持されているグローバル ロックは、関連付けられたセッションが kill されたときに無効になります。ロックによって、ロックを保持しているセッションが、設定に対して排他的な書き込みアクセスを行えるようになります。グローバル ロックにより設定の変更が拒否された場合は、エラー メッセージによって、NETCONF グローバル ロックが原因で設定の変更が拒否されたことが示されます。

<lock> 操作は必須パラメータ <target> を受け取ります。これは、ロックしようとするコンフィギュレーション データストアの名前です。ロックがアクティブな場合、<edit-config> 操作と <copy-config> 操作は許可されません。

NETCONF のグローバル ロックの保持中に **clear configuration lock** コマンドが指定された場合は、設定の完全な同期がスケジュールされ、警告の syslog メッセージが生成されます。このコマンドは、パーサー コンフィギュレーション ロックのみをクリアします。

次に、<lock> 操作を示す RPC の例を示します。

```
<rpc message-id="101"
          xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
       <lock>
         <target>
           <running/>
```

```
        </target>
      </lock>
</rpc>
```

NETCONF Kill セッション

セッションの競合時、またはクライアントによるグローバル ロックの誤用が生じたときは、**show netconf-yang sessions** コマンドを使用して NETCONF セッションをモニタできます。また、**clear netconf-yang session** コマンドを使用して応答しなくなったセッションをクリアすることもできます。**clear netconf-yang session** コマンドは、NETCONF ロックとコンフィギュレーション ロックの両方をクリアします。

<kill-session> 要求は、NETCONF セッションを強制的に終了します。NETCONF エンティティは、オープン セッションの <kill-session> 要求を受信すると、プロセス内のすべての操作を停止し、セッションに関連付けられているすべてのロックとリソースを解放して、関連付けられた接続をすべて閉じます。

<kill-session> 要求には、終了する NETCONF セッションのセッション ID が必要です。セッション ID の値が現在のセッション ID と同じ場合は、無効な値を示すエラーが返されます。NETCONF セッションのトランザクションがまだ進行中に NETCONF セッションが終了した場合は、データ モデル インフラストラクチャによってロールバックが要求され、ネットワーク要素にロールバックが適用されて、すべての YANG モデルの同期がトリガーされます。

セッションの kill が失敗し、グローバル ロックが保持されている場合は、コンソールまたは vty を使用して **clear configuration lock** コマンドを入力します。この時点で、データ モデルを停止して再起動することができます。

NETCONF プロトコルの設定方法

NETCONF-YANG は、デバイスのプライマリ トラストポイントを使用します。トラストポイントが存在しない場合に NETCONF-YANG が設定されると、自己署名トラストポイントが作成されます。詳細については、『公開キー インフラストラクチャ コンフィギュレーション ガイド（Cisco IOS XE Gibraltar 16.10.x 向け）』を参照してください。

NETCONF を使用するための権限アクセスの提供

NETCONF API の使用を開始するには、権限レベル 15 を持つユーザである必要があります。そのようにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device# enable`	パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**username** *name* **privilege** *level* **password** *password* 例： `Device(config)# username example-name privilege 15 password example_password`	ユーザ名をベースとした認証システムを確立します。次のキーワードを設定します。 • **privilege** *level*：ユーザの権限レベルを設定します。プログラマビリティ機能の場合は、15 にする必要があります。 • **password** *password*：CLI ビューにアクセスするためのパスワードを設定します。

ステップ 4	**aaa authentication login default local** および **aaa authorization exec default local**
	例：
	`Device (config)# aaa authentication login default local`
	`Device (config)# aaa authorization exec default local`
		（任意）**aaa new-model** を設定する場合は、 AAA 認証および許可が必要です。リモート AAA サーバの場合は、local を AAA サーバに置き換えます。

ステップ 5	**end** 例： `Device (config)# end`	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。

NETCONF-YANG の設定

レガシー NETCONF プロトコルがデバイスで有効になっている場合、RFC 準拠の NETCONF プロトコルは機能しません。**no netconf legacy** コマンドを使用してレガシー NETCONF プロトコルを無効にしてください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**netconf-yang** 例： `Device (config)# netconf-yang`	ネットワーク デバイスで NETCONF インターフェイスを有効にします。 （注） CLI による最初のイネーブル化の後、ネットワーク デバイスをモデル ベースのインターフェイスを通じて管理できるようになります。モデル ベースのインターフェイス プロセスの完全なアクティベーションには、最大 90 秒かかることがあります。
ステップ 4	**netconf-yang feature candidate-datastore** 例： `Device(config)# netconf-yang feature candidate-datastore`	候補データストアを有効にします。
ステップ 5	**exit** 例： `Device (config)# exit`	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。

NETCONF オプションの設定

SNMP の設定

NETCONF を有効にして、サポートされている MIB から生成された YANG モデルを使用して SNMP MIB データにアクセスしたり、IOS でサポートされている SNMP トラップを有効にして、サポートされているトラップから NETCONF 通知を受信するには、IOS で SNMP サーバを有効にします。

次の操作を行ってください。

手順

ステップ 1 IOS で SNMP 機能を有効にします。

例：

```
configure terminal
logging history debugging
logging snmp-trap emergencies
logging snmp-trap alerts
logging snmp-trap critical
logging snmp-trap errors
logging snmp-trap warnings
logging snmp-trap notifications
logging snmp-trap informational
logging snmp-trap debugging
!
snmp-server community public RW
snmp-server trap link ietf
snmp-server enable traps snmp authentication linkdown linkup
snmp-server enable traps syslog
snmp-server manager
exit
```

ステップ 2 NETCONF-YANG が起動した後、次の RPC <edit-config> メッセージを NETCONF-YANG ポートに送信して、SNMP トラップのサポートを有効にします。

例：

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <netconf-yang xmlns="http://cisco.com/yang/cisco-self-mgmt">
        <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
          <snmp-trap-control>
            <trap-list>
              <trap-oid>1.3.6.1.4.1.9.9.41.2.0.1</trap-oid>
            </trap-list>
            <trap-list>
              <trap-oid>1.3.6.1.6.3.1.1.5.3</trap-oid>
            </trap-list>
            <trap-list>
              <trap-oid>1.3.6.1.6.3.1.1.5.4</trap-oid>
            </trap-list>
          </snmp-trap-control>
        </cisco-ia>
      </netconf-yang>
    </config>
  </edit-config>
</rpc>
```

ステップ 3 次の RPC メッセージを NETCONF-YANG ポートに送信して、実行コンフィギュレーションをスタートアップ コンフィギュレーションに保存します。

例：

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <cisco-ia:save-config xmlns:cisco-ia="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

NETCONF プロトコルのコンフィギュレーションの確認

NETCONF コンフィギュレーションを確認するには次のコマンドを使用します。

手順

ステップ 1 **show netconf-yang datastores**

NETCONF-YANG データストアに関する情報を表示します。

例：

```
Device# show netconf-yang datastores

Device# show netconf-yang datastores
Datastore Name : running
Globally Locked By Session : 42
Globally Locked Time : 2018-01-15T14:25:14-05:00
```

ステップ 2 **show netconf-yang sessions**

NETCONF-YANG セッションに関する情報を表示します。

例：

```
Device# show netconf-yang sessions

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore
Number of sessions : 10
session-id transport username source-host global-lock
--------------------------------------------------------------------------------
40 netconf-ssh admin 10.85.70.224 None
42 netconf-ssh admin 10.85.70.224 None
44 netconf-ssh admin 10.85.70.224 None
46 netconf-ssh admin 10.85.70.224 None
48 netconf-ssh admin 10.85.70.224 None
50 netconf-ssh admin 10.85.70.224 None
52 netconf-ssh admin 10.85.70.224 None
54 netconf-ssh admin 10.85.70.224 None
56 netconf-ssh admin 10.85.70.224 None
58 netconf-ssh admin 10.85.70.224 None
```

ステップ 3 **show netconf-yang sessions detail**

NETCONF-YANG セッションに関する詳細情報を表示します。

例：

```
Device# show netconf-yang sessions detail

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions     : 1
```

```
session-id             : 19
transport              : netconf-ssh
username               : admin
source-host            : 2001:db8::1
login-time             : 2018-10-26T12:37:22+00:00
in-rpcs                : 0
in-bad-rpcs            : 0
out-rpc-errors         : 0
out-notifications      : 0
global-lock            : None
```

ステップ 4 show netconf-yang statistics

NETCONF-YANG 統計に関する情報を表示します。

例：

```
Device# show netconf-yang statistics

netconf-start-time : 2018-01-15T12:51:14-05:00
in-rpcs : 0
in-bad-rpcs : 0
out-rpc-errors : 0
out-notifications : 0
in-sessions : 10
dropped-sessions : 0
in-bad-hellos : 0
```

ステップ 5 show platform software yang-management process

NETCONF-YANG のサポートに必要なソフトウェア プロセスのステータスを表示します。

例：

```
Device# show platform software yang-management process

confd            : Running
nesd             : Running
syncfd           : Running
ncsshd           : Running
dmiauthd         : Running
vtyserverutild   : Running
opdatamgrd       : Running
nginx            : Running
ndbmand          : Running
```

（注） プロセス nginx は、**ip http secure-server** または **ip http server** がデバイスで設定されている場合に実行されます。このプロセスが「実行」状態でなくても NETCONF は正常に機能します。ただし、RESTCONF には nginx プロセスが必要です。

表 *13 : show platform software yang-management process* のフィールドの説明

フィールド	説明
confd	コンフィギュレーション デーモン

フィールド	説明
nesd	ネットワーク要素シンクロナイザ デーモン
syncfd	デーモンからの同期
ncsshd	NETCONF セキュア シェル（SSH）デーモン
dmiauthd	デバイス管理インターフェイス（DMI）認証デーモン
vtyserverutild	VTY サーバ ユーティリティ デーモン
opdatamgrd	運用データ マネージャ デーモン
nginx	NGINX Web サーバ
ndbmand	NETCONF データベース マネージャ

NETCONF プロトコルの関連資料

関連資料

関連項目	マニュアル タイトル
IOS-XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースの YANG データ モデル	開発者に分かりやすい方法で Cisco YANG モデルにアクセスするには、*GitHub リポジトリ*を複製し、*vendor/cisco* サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

標準および RFC

標準/RFC	タイトル
RFC 6020	*YANG：Network Configuration Protocol（NETCONF）向けデータ モデリング言語*
RFC 6241	ネットワーク設定プロトコル（*NETCONF*）
RFC 6536	ネットワーク設定プロトコル（*NETCONF*）アクセス制御モデル
RFC 8040	*RESTCONF プロトコル*

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

NETCONF プロトコルの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

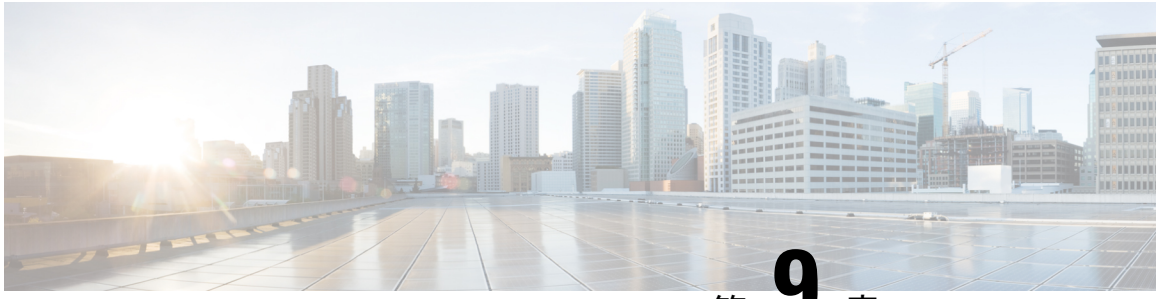
表 *14 : NETCONF* プロトコルの機能情報

機能名	リリース	機能情報
候補コンフィギュレーション サポート	Cisco IOS XE Fuji 16.9.1	候補コンフィギュレーション サポート機能を使用すると、シンプルなコミットオプションを使用して RFC 6241 を実装することによって、候補機能をサポートできます。 この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco ISR 4000 シリーズ サービス統合型ルータ 次のコマンドが導入されました： **netconf-yang feature candidate-datastore**

機能名	リリース	機能情報
NETCONF プロトコル	Cisco IOS XE Denali 16.3.1	NETCONFプロトコル機能によって、プログラムによる各種の標準規格に準拠した方法で、設定の記述やネットワーク デバイスからの運用データの読み取りが容易になります。 次のコマンドが導入されました：**netconf-yang**
	Cisco IOS XE Everest 16.5.1a	この機能は、Cisco Catalyst 9300 シリーズ スイッチと Cisco Catalyst 9500 シリーズ スイッチに実装されました。
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco Catalyst 9400 シリーズ スイッチに実装されました。
	Cisco IOS XE Fuji 16.7.1	この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Fuji 16.8.1a	この機能は、Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。

機能名	リリース	機能情報
NETCONF および RESTCONF IPv6 のサポート	Cisco IOS XE Fuji 16.8.1a	この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco CSR 1000v スイッチ • Cisco ISR 1100 シリーズ サービス統合型ルータ • Cisco ISR 4000 シリーズ サービス統合型ルータ

機能名	リリース	機能情報
NETCONF グローバル ロックおよびセッションの kill	Cisco IOS XE Fuji 16.8.1a	この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco CSR 1000v スイッチ • Cisco ISR 1100 シリーズ サービス統合型ルータ • Cisco ISR 4000 シリーズ サービス統合型ルータ



第 9 章

RESTCONF プロトコル

この章では、HTTP ベースの Representational State Transfer コンフィギュレーション プロトコル（RESTCONF）を設定する方法を説明します。RESTCONF は、設定データ、状態データ、データ モデルに固有のリモート プロシージャ コール（RPC）操作、および YANG モデルで定義されているイベントにアクセスするための、標準的なメカニズムに基づく、プログラミングが可能なインターフェイスを提供します。

RESTCONF プロトコルの前提条件

- RESTCONF に対して Cisco IOS-HTTP サービスを有効にします。詳細については、『RESTCONF RPC の例』を参照してください。

RESTCONF プロトコルの制約事項

RESTCONF プロトコルには、次の制約事項が適用されます。

- 通知およびイベント ストリーム
- YANG パッチ
- フィルタ、開始時、停止時、再生、アクションなどのオプションのクエリ パラメータ
- RESTCONF 機能は、デュアル IOSd 設定またはソフトウェア冗長性を実行しているデバイスではサポートされていません。

RESTCONF プログラマブル インターフェイスについて

RESTCONF の概要

このセクションでは、構成をネットワーク デバイスにプログラムを使用して書き込めるようにする、プロトコルおよびモデリング言語について説明します。

- RESTCONF：構造化データ（XML または JSON）および YANG を使用して REST ライクな API を提供します。これによりさまざまなネットワーク デバイスにプログラムを使用してアクセスできます。RESTCONF API は HTTPs メソッドを使用します。
- YANG：モデル構成および操作機能に使用されるデータ モデリング言語。YANG は、NETCONF および RESTCONF API によって実行できる関数の有効範囲と種類を決定します。

IOS での RESTCONF および NETCONF

プログラマチック デバイスのプロトコルおよびデータ モデル

このセクションでは、構成をネットワーク デバイスにプログラムを使用して書き込めるようにする、プロトコルおよびモデリング言語について説明します。

- RESTCONF：構造化データ（XML または JSON）および YANG を使用して REST ライクな API を提供します。これによりさまざまなネットワーク デバイスにプログラムを使用してアクセスできます。RESTCONF API は HTTPs メソッドを使用します。
- YANG：モデル構成および操作機能に使用されるデータ モデリング言語。YANG は、NETCONF および RESTCONF API によって実行できる関数の有効範囲と種類を決定します。

RESTCONF サーバが NETCONF サーバと共存している場合、NETCONF プロトコルとのプロトコル インタラクションがあります。RESTCONF サーバは、操作リソースを使用して特定のデータストアへのアクセスを提供します。ただし RESTCONF プロトコルは必須の操作リソースを指定していないので、各操作リソースはデータストアにアクセスするかどうか、およびその方法を決定します。

詳細については、『Catalyst 4500 Series Software Configuration Guide』の「Protocols and Data Models for Programmatic Device」のセクションを参照してください。

HTTPs メソッド

ステートレス プロトコルである https ベースのプロトコル RESTCONF（RFC 8040）は、セキュアな HTTP メソッドを使用して、YANG 定義データが含まれる概念データストア（NETCONF データストアを実装するサーバと互換性がある）で CREATE、READ、UPDATE、および DELETE（CRUD）操作を提供します。

次の表では、RESTCONF 操作に NETCONF プロトコル操作を関連付ける方法を示しています。

オプション	サポートされているメソッド
GET	読み取り
PATCH	更新
PUT	作成または置換
POST	作成または操作（リロード、デフォルト）
DELETE	ターゲット リソースの削除
HEAD	ヘッダー メタデータ（応答本文なし）

RESTCONF ルート リソース

- RESTCONF デバイスは、RESTCONF 属性を含むリンク要素である /.well-known/host-meta リソースにより、RESTCONF API のルートを決定します。
- RESTCONF デバイスは、要求 URI のパスの最初の部分として restconf API ルート リソースを使用します。

次に例を示します。

```
Example returning /restconf:

   The client might send the following:

      GET /.well-known/host-meta HTTP/1.1
      Host: example.com
      Accept: application/xrd+xml

   The server might respond as follows:

      HTTP/1.1 200 OK
      Content-Type: application/xrd+xml
      Content-Length: nnn

      <XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
          <Link rel='restconf' href='/restconf'/>
      </XRD>
```

URI の例：

- GigabitEthernet0/0/2：
  http://10.104.50.97/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=0%2F0%2F2
- fields=name：
  http://10.104.50.97/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=0%2F0%2F2?fields=name
- depth=1：
  https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet?depth=1

- 名前と IP：
  https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native/interface?fields=GigabitEthernet/ip/address/primary;name
- MTU（フィールド）：
  https://10.104.50.97/restconf/data/Cisco-IOS-XE-native:native/interface?fields=GigabitEthernet(mtu)
- MTU：
  https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=3/mtu
- ポートチャネル：
  https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native/interface/Port-channel
- 「Char」から「Hex」への変換チャート：http://www.columbia.edu/kermit/ascii.html

RESTCONF API リソース

API リソースは、+restconfに位置する上位リソースです。これは次のメディア タイプをサポートします。

- application/yang-data+xml または application/yang-data+json
- API リソースには、RESTCONF DATASTORE および OPERATION リソースの RESTCONF ルート リソースが含まれます。次に例を示します。

```
The client may then retrieve the top-level API resource, using the
   root resource "/restconf".

      GET /restconf HTTP/1.1
      Host: example.com
      Accept: application/yang-data+json

   The server might respond as follows:

      HTTP/1.1 200 OK
      Date: Thu, 26 Jan 2017 20:56:30 GMT
      Server: example-server
      Content-Type: application/yang-data+json

      {
        "ietf-restconf:restconf" : {
          "data" : {},
          "operations" : {},
          "yang-library-version" : "2016-06-21"
        }
      }
```

詳細については、RFC 3986 を参照してください

予約文字または予約されていない文字

Conbody

メソッド

コンテンツ クエリ パラメータは、要求されたデータ ノードの子孫ノードが応答でどのように処理されるかを制御します。

- サーバによってサポートされている必要があります。
- URI 内に存在しない場合のデフォルト値は、all です。GET/HEAD メソッドに対してのみ許可されます。

  「400 の不正要求」ステータス行は、他のメソッドまたはリソース タイプに使用される場合に返されます。

許可される値の例は次のとおりです。

1. https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native?content=config
2. https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native?content=nonconfig’

クエリ パラメータ（フィールド）

- depth-queryパラメータは、サーバによって返されるサブツリーの深さを制限するために使用されます。
- 「depth」パラメータの値は 1 ～ 65535 の整数またはストリング「unbounded」のいずれかです。
- 機能 URI に存在する場合にサポートされます。
- URI 内に存在しない場合のデフォルト値は、「unbounded」です。
- GET/HEAD メソッドに対してのみ許可されます。

  「400 の不正要求」ステータス行は、他のメソッドまたはリソース タイプに使用される場合に返されます。

例：

```
1) ‘https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native?content=config&depth=65535’

2) ‘https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native?content=nonconfig&depth=0’

>>> resp

<Response [400]>

>>> resp.text

'{"errors": {"error": [{"error-message": "invalid value for depth query parameter",
"error-tag": "malformed-message", "error-type": "application"}]}}\n'

>>>
```

例：

- 「フィールド」クエリ パラメータは、GET メソッドで取得される、ターゲット リソース内のデータ ノードを識別するためにオプションで使用されます。

- 機能 URI に存在する場合にサポートされます。

  GET/HEAD メソッドに対してのみ許可されます。

- 「400 の不正要求」ステータス行は、他のメソッドまたはリソース タイプに使用される場合に返されます。

- 「フィールド」クエリ パラメータの値は、次のルールと一致します。

```
fields-expr = path "(" fields-expr ")" / path ";" fields-expr / path path =
api-identifier [ "/" path ]
```

- **1.** 複数のノードを選択するには、「;」を使用します。

  2. ノードのサブセレクタを指定するには、かっこを使用します。「path」フィールドと左かっこ文字「(」の間にパス区切り文字「/」がないことに注意してください。

  3. 「/」は、パス内でノードの子ノードを取得するために使用します。

- 「フィールド」クエリ パラメータの値は、次のルールと一致します。

```
fields-expr = path "(" fields-expr ")" / path ";" fields-expr / path path =
api-identifier [ "/" path ]
```

- **1.** 複数のノードを選択するには、「;」を使用します。

  2. ノードのサブセレクタを指定するには、かっこを使用します。「path」フィールドと左かっこ文字「(」の間にパス区切り文字「/」がないことに注意してください。

  3. 「/」は、パス内でノードの子ノードを取得するために使用します。

例：

1. サーバ モジュールの情報：'https://10.85.116.59:443/restconf/data?fields=ietf-yang-library:modules-state/module(name;revision;schema;namespace)'

2. 名前と IP：'https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native/interface?fields=GigabitEthernet/ip/address/primary;name'

クエリ パラメータ（ポイント）

- 「ポイント」クエリ パラメータは、順序ユーザ リストまたはリーフリスト内で作成されたり移動したりするデータ リソースの挿入ポイントを指定するために使用されます。

- サーバによってサポートされている必要があります。

  - POST および PUT メソッドにのみ許可されます。

    「ポイント」のパラメータの値は、挿入ポイント オブジェクトへのパスを識別する文字列です。その形式は、ターゲット リソース URI 文字列と同じです。

    例：

```
PUT:
'https://10.85.116.59:443/restconf/data/Cisco-IOS-XE-native:native/privilege/exec/level=2/cmd=test-show2&ins=after&point=%2Fcisco-IOS-XE-native%3Anative%2Fprivilege%2Fexec%2Flevel%3D2%2Fcmd%3Dtest-show2'
```

```
{
  "Cisco-IOS-XE-native:command-list": [
    {
      "command": "show terminal"
    }
  ]
```

クエリ パラメータ（デフォルトあり）

「デフォルトあり」クエリ パラメータは、デフォルトのデータ ノードに関する情報が、データ リソースに対する GET 要求への応答でどのように返されるかを指定します。機能のデフォルトの基本モードは明示的です。

値	説明
Report-All	すべてのデータ ノードが報告されます。
Trim	YANG のデフォルトに設定されたデータ ノードは報告されません。
Explicit	クライアントにより YANG のデフォルトに設定されたデータ ノードが報告されます。

- 「ポイント」クエリ パラメータは、順序ユーザ リストまたはリーフリスト内で作成されたり移動したりするデータ リソースの挿入ポイントを指定するために使用されます。

  例：

```
Sync default settings (error):
'https://10.85.116.59:443/restconf/data/cisco-self-mgmt:netconf-yang/cisco-ia:cisco-ia/cisco-ia:logging/cisco-ia:sync-log-level?with-defaults=report-all'
Intelligent sync (true):
'https://10.85.116.59:443/restconf/data/cisco-self-mgmt:netconf-yang/cisco-ia:cisco-ia/cisco-ia:intelligent-sync?with-defaults=report-all'
```

RESTCONF プログラマブル インターフェイスの設定方法

AAA を使用した NETCONF/RESTCONF の認証

始める前に

NETCONF 接続と RESTCONF 接続は、認証、許可、およびアカウンティング（AAA）を使用して認証する必要があります。その結果、権限レベル 15 のアクセスで定義された RADIUS または TACACS + ユーザに、システムへのアクセスが許可されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa group server radius** *server-name* 例： `Device(config)# aaa group server radius ISE`	RADIUS サーバを追加し、サーバ グループ RADIUS コンフィギュレーション モードを開始します。 • server-name 引数には、RADIUS サーバ グループ名を指定します。
ステップ 5	**server-private** *ip-address* **key** *key-name* 例： `Device(config-sg-radius)# server-private 172.25.73.76 key Cisco123`	プライベート RADIUS サーバの IP アドレスと暗号キーを設定します。
ステップ 6	**ip vrf forwarding** *vrf-name* 例： `Device(config-sg-radius)# ip vrf forwarding Mgmt-intf`	AAA RADIUS または TACACS+ サーバ グループの Virtual Route Forwarding（VRF）参照情報を設定します。
ステップ 7	**exit** 例： `Device(config-sg-radius)# exit`	サーバ グループ RADIUS 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 8	**aaa authentication login default group** *group-name***local** 例： `Device(config)# aaa authentication login default group ISE local`	ログイン時に、指定されたグループ名をデフォルトのローカル AAA 認証として設定します。

	コマンドまたはアクション	目的
ステップ 9	**aaa authentication login** *list-name* **none** 例： `Device(config)# aaa authentication login NOAUTH none`	システムへのログイン中に認証が不要であることを指定します。
ステップ 10	**aaa authorization exec default group** *group-name***local** 例： `Device(config)# aaa authorization exec default group ISE local`	許可を実行して、EXEC シェルの実行がユーザに許可されているかどうかを確認します。
ステップ 11	**aaa session-id common** 例： `Device(config)# aaa session-id common`	指定のコールに対して送信されたセッション ID 情報が同じになるようにします。
ステップ 12	**line console** *number* 例： `Device(config)# line console 0`	設定する特定の回線を識別し、ラインコンフィギュレーションモードを開始します。
ステップ 13	**login authentication** *authentication-list* 例： `Device(config-line)# login authentication NOAUTH`	ログインに対する AAA 認証をイネーブルにします。
ステップ 14	**end** 例： `Device(config-line)# end`	回線コンフィギュレーションモードを終了します。続いて、特権 EXEC モードに戻ります。

RESTCONF の Cisco IOS HTTP サービスの有効化

RESTCONF インターフェイスを使用するには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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。

	コマンドまたはアクション	目的
ステップ 3	**restconf** 例： `Device(config)# restconf`	ネットワーク デバイスで RESTCONF インターフェイスを有効にします。
ステップ 4	**ip http secure-server** 例： `Device(config)# ip http secure-server`	セキュア HTTP（HTTPS）サーバをイネーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

RESTCONF の設定の検証

スタートアップ コンフィギュレーションを使用してデバイスが起動すると、*nginx* プロセスが実行中になります。ただし、DMI プロセスは有効にはなりません。

次の **show platform software yang-management process monitor** コマンドの出力例は、nginx プロセスが実行中であることを示しています。

```
Device# show platform software yang-management process monitor

COMMAND           PID S    VSZ   RSS %CPU %MEM     ELAPSED
nginx           27026 S 332356 18428  0.0  0.4       01:34
nginx           27032 S 337852 13600  0.0  0.3       01:34
```

NGINX は、プロキシ Web サーバとして機能する内部 Web サーバで、Transport Layer Security（TLS）ベースの HTTPS を提供します。HTTPS を介して送信された RESTCONF 要求は、最初に NGINX プロキシ Web サービスによって受信され、さらに要求が構文/セマンティックチェックのために confd Web サーバに転送されます。

次の **show platform software yang-management process** コマンドの出力例は、スタートアップ コンフィギュレーションを使用してデバイスが起動されたときのすべてのプロセスのステータスを示しています。

```
Device# show platform software yang-management process

confd            : Not Running
nesd             : Not Running
syncfd           : Not Running
ncsshd           : Not Running
dmiauthd         : Not Running
nginx            : Running
ndbmand          : Not Running
pubd             : Not Running
```

restconf コマンドが設定されている場合、nginx プロセスが再起動され、DMI プロセスが起動されます。

次の **show platform software yang-management process** コマンドの出力例は、nginx プロセス と DMI プロセスが起動して実行中であることを示しています。

```
Device# show platform software yang-management process

confd            : Running
nesd             : Running
syncfd           : Running
ncsshd           : Not Running ! NETCONF-YANG is not configured, hence ncsshd process
is in not running.
dmiauthd         : Running
vtyserverutild   : Running
opdatamgrd       : Running
nginx            : Running ! nginx process is up due to the HTTP configuration, and it
is restarted when RESTCONF is enabled.
ndbmand          : Running
```

次の show platform software yang-management process monitor コマンドの出力例では、すべてのプロセスに関する詳細情報が表示されています。

```
Device#show platform software yang-management process monitor

COMMAND           PID S    VSZ   RSS %CPU %MEM     ELAPSED
confd           28728 S 860396 168496 42.2  4.2       00:12
confd-startup.s 28448 S  19664   4496  0.2  0.1       00:12
dmiauthd        29499 S 275356 23340  0.2  0.5       00:10
ndbmand         29321 S 567232 65564  2.1  1.6       00:11
nesd            29029 S 189952 14224  0.1  0.3       00:11
nginx           29711 S 332288 18420  0.6  0.4       00:09
nginx           29717 S 337636 12216  0.0  0.3       00:09
pubd            28237 S 631848 68624  2.1  1.7       00:13
syncfd          28776 S 189656 16744  0.2  0.4       00:12
```

AAA と RESTCONF インターフェイスが設定され、nginx プロセスと関連する DMI プロセスが実行中になった後、デバイスは RESTCONF 要求を受信できる状態になります。

NETCONF/RESTCONF セッションのステータスを表示するには、**show netconf-yang sessions** コマンドを使用します。

```
Device# show netconf-yang sessions

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions : 1

session-id  transport    username             source-host            global-lock
--------------------------------------------------------------------------------
19          netconf-ssh  admin                2001:db8::1             None
```

NETCONF/RESTCONF セッションに関する詳細情報を表示するには、**show netconf-yang sessions detail** コマンドを使用します。

```
Device# show netconf-yang sessions detail

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions     : 1

session-id             : 19
transport              : netconf-ssh
username               : admin
source-host            : 2001:db8::1
login-time             : 2018-10-26T12:37:22+00:00
in-rpcs                : 0
in-bad-rpcs            : 0
out-rpc-errors         : 0
out-notifications      : 0
global-lock            : None
```

RESTCONF プログラマブル インターフェイスの設定例

例：RESTCONF プロトコルの設定

RESTCONF 要求（HTTPS Verb）：

次に、ターゲット リソースで許可されている HTTPS Verb を示す RESTCONF 要求の例を示します。この例では **logging monitor** コマンドを使用しています。

```
root:~# curl -i -k -X "OPTIONS"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin'
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 23 Apr 2018 15:27:57 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Allow: DELETE, GET, HEAD, PATCH, POST, PUT, OPTIONS    >>>>>>>>>>>    Allowed methods
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Accept-Patch: application/yang-data+xml, application/yang-data+json
Pragma: no-cache

root:~#
```

POST（作成）要求

POST 操作では、ターゲット デバイスに存在しないコンフィギュレーションが作成されます。

（注） 実行コンフィギュレーションで **logging monitor** コマンドを使用できないことを確認してください。

次の POST 要求の例では **logging monitor alerts** コマンドを使用しています。

```
Device:~# curl -i -k -X "POST"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor" \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin' \
>      -d $'{
>   "severity": "alerts"
> }'
HTTP/1.1 201 Created
Server: nginx
Date: Mon, 23 Apr 2018 14:53:51 GMT
Content-Type: text/html
Content-Length: 0
Location:
https://10.85.116.30/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 14:53:51 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-495231-97239
Pragma: no-cache

Device:~#
```

PUT：（作成または置換）要求：

指定されたコマンドがデバイスに存在しない場合は、POST 要求によって作成されます。ただし、実行コンフィギュレーションにすでに存在する場合は、この要求によってコマンドが置き換えられます。

次の PUT 要求の例では **logging monitor warnings** コマンドを使用しています。

```
Device:~# curl -i -k -X "PUT"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin' \
>      -d $'{
>   "severity": "warnings"
> }'
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Apr 2018 14:58:36 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 14:57:46 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-495466-326956
Pragma: no-cache

Device:~#
```

PATCH：（更新）要求

次の PATCH 要求の例では **logging monitor informational** コマンドを使用しています。

```
Device:~# curl -i -k -X "PATCH"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native" \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin' \
>      -d $'{
>   "native": {
>     "logging": {
>       "monitor": {
>         "severity": "informational"
>       }
>     }
>   }
> }'
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Apr 2018 15:07:56 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 15:07:56 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-496076-273016
Pragma: no-cache
Device:~#
```

GET 要求（読み取り）

次の GET 要求の例では **logging monitor informational** コマンドを使用しています。

```
Device:~# curl -i -k -X "GET"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin'
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 23 Apr 2018 15:10:59 GMT
Content-Type: application/yang-data+json
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Pragma: no-cache

{
  "Cisco-IOS-XE-native:severity": "informational"
}
Device:~#
```

DELETE 要求（コンフィギュレーションの作成）

```
Device:~# curl -i -k -X "DELETE"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin'
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Apr 2018 15:26:05 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 15:26:05 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-497165-473206
Pragma: no-cache

linux_host:~#
```

RESTCONF プロトコルの関連資料

関連資料

関連項目	マニュアル タイトル
IOS-XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースの YANG データ モデル	開発者にわかりやすい方法で Cisco YANG モデルにアクセスするには、GitHub リポジトリを複製し、vendor/cisco サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

標準および RFC

標準/RFC	タイトル
RFC 6020	YANG：Network Configuration Protocol（NETCONF）向けデータ モデリング言語
RFC 8040	Representational State Transfer Configuration Protocol（RESTCONF）

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	https://www.cisco.com/c/en/us/support/index.html

RESTCONF プロトコルの機能情報

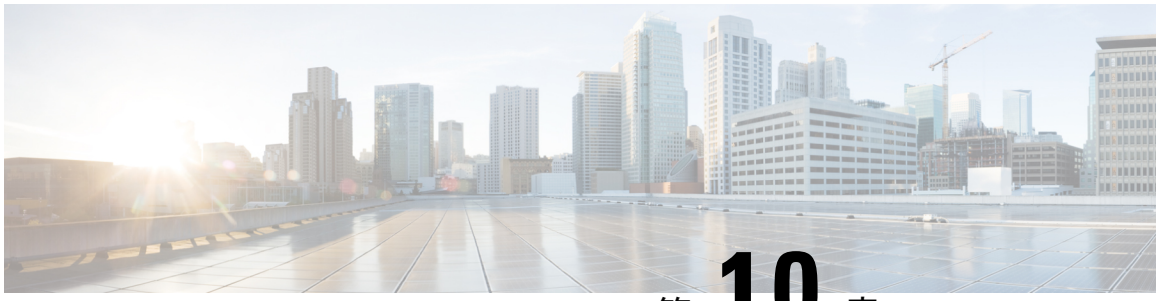
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

表 *15 : RESTCONF* プロトコルの機能情報

機能名	リリース	機能情報
RESTCONF プロトコル	Cisco IOS XE Everest 16.6.1	この章では、HTTP ベースのプロトコルである Representational State Transfer Configuration Protocol（RESTCONF）の設定および構成方法を説明します。RESTCONF は、YANG モデルで定義されている設定データ、状態データ、データ モデル固有のリモート プロシージャ コール（RPC）の操作およびイベント通知にアクセスするための、標準メカニズムに基づくプログラマチック インターフェイスを提供します。 この機能は、ASR 1000 アグリゲーション サービス ルータの ASR1001-HX および ASR1002-HX、CSR 1000v シリーズ クラウド サービス ルータ、および Cisco 4000 シリーズ サービス統合型ルータ（ISR）に導入されました。 次のコマンドが導入または変更されました：**ip http server** および **restconf**

機能名	リリース	機能情報
	Cisco IOS XE Fuji 16.8.1a	この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ



第 10 章

運用データ パーサーのポーリング

YANG データ モデルでは、デバイスからの運用状態データを読み取ることができます。

運用データ パーサーのポーリングについて

運用データの概要

YANGデータ モデルを使用すると、デバイスから運用状態データを読み取ることができます。運用データでは、IOS **show** コマンドと同様に、デバイスの現在の状態や動作を判定することができます。

読み取り専用の運用状態データをシステムから取得するには、NETCONF GET 操作を実行します。適切なYANGモデルを介してデータを取得するには、NETCONF を有効にし、（該当する場合）データ パーサーをアクティブ化する必要があります。

プログラミングが可能なインターフェイスとCLIを介して運用データを設定する方法の詳細については、「運用データの設定方法」の項を参照してください。

運用データ パーサーと対応する YANG モデル

運用データ パーサーには、2 つのタイプがあります。1 つ目のタイプは常にオンにするものです。2 つ目のタイプは、運用データを一定の間隔でポーリングするように設定する必要があります。1つ目のタイプの運用データ パーサーについては、設定は不要です。データは、NETCONF GET 要求の際に常に取得されます。これらのデータ パーサーにはポーリング間隔はなく、運用データは、運用データは変更が発生するとすぐに更新されます。

2 つ目のタイプの運用データ パーサーは、CLI または NETCONF メッセージを介してアクティブ化する必要があります（詳細については、「運用データ パーサーのポーリングを有効にする

方法」の項を参照してください）。このタイプのパーサーの運用データは定期的なポーリング間隔でポーリングされ、その情報は、NETCONF GET 要求の際に取得されます。

次の表に、アクティブ化が必要なデータ パーサーと、それらに対応する、運用データが格納されている YANG モデルを示します。

表 *16* : アクティブ化が必要な運用データ パーサーと対応する *YANG* モデル

運用データ パーサー名	運用データにアクセスする YANG モデル
BGP	Cisco-IOS-XE-bgp-oper.yang
BFD	Cisco-IOS-XE-bfd-oper.yang
BridgeDomain	Cisco-IOS-XE-bridge-domain.yang （注） ルーティング プラットフォームでのみサポート
DiffServ	ietf-diffserv-target.yang
EthernetCFMStats	Cisco-IOS-XE-cfm-oper.yang （注） ルーティング プラットフォームでのみサポート
FlowMonitor	Cisco-IOS-XE-flow-monitor-oper.yang
IPRoute	ietf-routing.yang
MPLSLForwarding	Cisco-IOS-XE-mpls-fwd-oper.yang
MPLSLDPNeighbor	Cisco-IOS-XE-mpls-ldp.yang
MPLSStaticBinding	common-mpls-static.yang
OSPF	ietf-ospf.yang
PlatformSoftware	Cisco-IOS-XE-platform-software-oper.yang
VirtualService	Cisco-IOS-XE-virtual-service-oper.yang （注） ルーティング プラットフォームでのみサポート

運用データ パーサーのポーリングを有効にする方法

プログラマブル インターフェイスを使用しての運用データ パーサー ポーリングの有効化

プログラマブル インターフェイスを使用して運用データ パーサーのポーリングを有効化するには、次の作業を実行します。

1. NETCONF-YANG を有効化した後に、cisco-odm.yang（GitHub リポジトリにて入手可能）を使用して <edit-config> リモート プロシージャ コール（RPC）を送信し、運用データのポーリングを有効化します。このポーリングを有効にすると、すべての運用データ パーサーがデフォルトで有効化します。各パーサーのデフォルトのポーリング間隔は、120 秒（120000 ミリ秒）です。ポーリング間隔により、パーサーが運用データを取得してデータストア内の対応する YANG モデルを更新する頻度が決定されます。
2. 運用データのポーリングを有効化したら、<get> RPC を送信し、運用データを取得します。運用データの取得にどの運用 YANG モデルを使用する必要があるか決定するには、パーサーから YANG モデルへのマッピングを使用します。次の RPC 応答は、Cisco-IOS-XE-acl-oper.yang を使用して、アクセス制御リスト（ACL）の運用データを取得します。

```
     CORRESPONDING RPC REPLY:
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <data>
        <access-lists xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-acl-oper">
            <access-list>
                <access-control-list-name>TEST</access-control-list-name>
                <access-list-entries>
                    <access-list-entry>
                        <rule-name>10</rule-name>
                        <access-list-entries-oper-data>
                        <match-counter>100</match-counter>
                    </access-list-entry>
                    <access-list-entry>
                        <rule-name>20</rule-name>
                        <access-list-entries-oper-data>
                        <match-counter>122</match-counter>
                    </access-list-entry>
                </access-list-entries>
            </access-list>
        </access-lists>
    </data>
</rpc-reply>
```

（注） 詳細については、*GitHub* リポジトリ内の cisco odm.yang モデルを参照してください。

CLI からの運用データ パーサーのポーリングの有効化

NETCONF-YANG を有効にしたら、このタスクを実行して運用データ パーサーのポーリングを有効にし、ポーリング間隔を調整します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	• パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**netconf-yang cisco-odm polling-enable** 例： `Device(config)# netconf-yang cisco-odm polling-enable`	運用データのポーリングを有効にします。
ステップ 4	**netconf-yang cisco-odm actions** *action-name* 例： `Device(config)# netconf-yang cisco-odm actions OSPF`	指定されたアクションを有効にし、ODM アクション コンフィギュレーション モードを開始します。 • 運用データを取得する運用データ パーサー名を指定します。
ステップ 5	**mode poll** 例： `Device(config-odm-action)# mode poll`	ポーリング モードでデータ パーサーを設定します。
ステップ 6	**polling-interval** *seconds* 例： `Device(config-odm-action)# polling-interval 1000`	デフォルトのパーサー ポーリング間隔を変更します。 • データのポーリングからパーサーを停止するには、**mode none** コマンドを設定します。
ステップ 7	**end** 例： `Device(config-odm-action)# end`	ODM アクション コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

運用データのポーリングを有効にしたら、<get> RPC を送信して、デバイスから運用データを入手します。

運用データパーサーのポーリングに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS XE の YANG データ モデル	開発者に分かりやすい方法でCisco YANGモデルにアクセスするには、*GitHub* リポジトリを複製し、*vendor/cisco* サブディレクトリに移動します。
	『Programmability Command Reference, Cisco IOS XE Everest 16.6.1』

MIB

MB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートWebサイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

運用データ パーサーのポーリングの機能情報

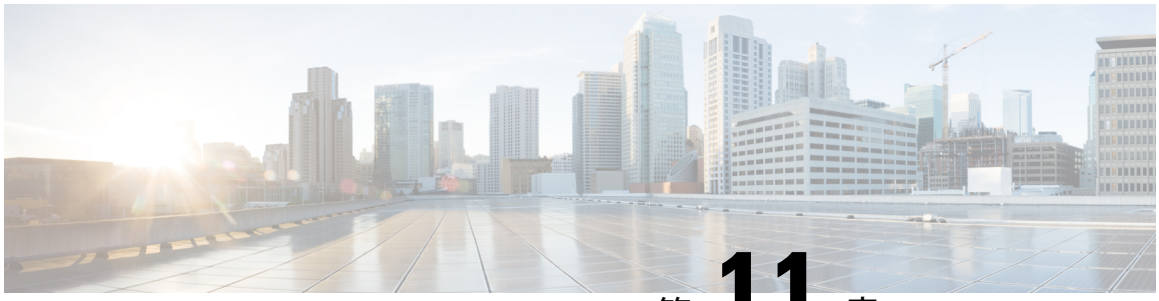
次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだ

けを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *17*：運用データ パーサーのポーリングの機能情報

機能名	リリース	機能情報
運用データ パーサーのポーリング	Cisco IOS XE Denali 16.3.1	YANG データ モデルにより、デバイスから運用状態データを読み取ることができます。
	Cisco IOS XE Everest 16.5.1a	この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ



第 11 章

モデル駆動型テレメトリ

モデル駆動型テレメトリ

モデル駆動型テレメトリでは、ネットワークデバイスからサブスクライバに、リアルタイムの設定や運用状態の情報を継続的にストリームすることができます。

アプリケーションは NETCONF-YANG により標準ベースの YANG データ モデルを使用して、必要とする特定のデータ項目をサブスクライブします。

構造化データは、サブスクリプション基準とデータ タイプに基づき、定義されたパターンでまたは変更時にパブリッシュされます。

モデル駆動型テレメトリの前提条件

- NETCONF-YANG の知識とその使用方法。これには以下が含まれます。
  - NETCONF セッションの確立。
  - hello メッセージと機能メッセージの送信および受信。
  - 確立された NETCONF セッションによる YANG XML リモート プロシージャ コール（RPC）の送受信詳細については、『NETCONF-YANG の設定例』を参照してください。

  NETCONF-YANG の詳細については、『データモデル』の章を参照してください。
- XML、XML 名前空間、および XML XPath の知識。
- IETF 動的テレメトリ仕様で定義された標準および原則の知識。
- NETCONF-YANG がデバイス上で設定済みであり稼働している必要があります。**show platform software yang-management process** コマンドを使用して、次のプロセスが実行中であることを確認します。

```
Device# show platform software yang-management process
```

```
confd            : Running
nesd             : Running
syncfd           : Running
ncsshd           : Running
dmiauthd         : Running
vtyserverutild   : Running
opdatamgrd       : Running
nginx            : Running
ndbmand          : Running
pubd             : Running
```

（注） プロセス pubd はモデル駆動型テレメトリ プロセスであり、これが実行していない場合にはモデル駆動型テレメトリは機能しません。

- *urn:ietf:params:netconf:capability:notification:1.1* 機能は、hello メッセージでリストする必要があります。この機能は、IETF テレメトリをサポートするデバイスでのみアドバタイズされます。

モデル駆動型テレメトリについて

モデル駆動型テレメトリの概要

テレメトリは、自動の通信プロセスです。これにより、測定およびその他のデータがリモートポイントまたはアクセス不能なポイントで収集され、モニタ用の受信装置に送信されます。モデル駆動型テレメトリは、モデル駆動型テレメトリ対応デバイスから送信先へとデータをストリームするメカニズムを提供します。

テレメトリでは、情報の送信元と送信先を識別するためにサブスクリプションモデルが使用されます。モデル駆動型テレメトリでは、ネットワーク要素の定期的なポーリングが不要になります。その代わりに、ネットワーク要素により、サブスクライバに配信される情報の継続的な要求が確立されます。その後、定期的に、またはオブジェクトの変更のたび、サブスクライブされているYANGオブジェクトのセットが当該のサブスクライバにストリーミングされます。

ストリーミングされるデータは、サブスクリプションによって駆動されます。サブスクリプションにより、アプリケーションはYANGデータストアから更新（自動または継続的な更新）をサブスクライブすることができるようになるため、発行者は、それらの更新をプッシュし、実質的にストリーミングすることができます。

サブスクリプションの概要

サブスクリプションは、テレメトリ ロール間の関連付けを作成する項目であり、ロール間で送信されるデータを定義します。

具体的には、サブスクリプションは、テレメトリデータの一部として要求される一連のデータを定義するために使用されます。たとえば、データがいつ必要か、データの書式設定の方法、また暗黙的でない場合は誰（どの受信者）がデータを受信するかを定義します。

サポートされているサブスクリプションの最大数はプラットフォームによって異なりますが、現在は100個のサブスクリプションがサポートされています。サブスクリプションは、設定済みか動的のいずれかにすることができ、トランスポートプロトコルの任意の組み合わせを使用できます。有効なすべての設定済みサブスクリプションをアクティブにするために同時に多数のサブスクリプションが動作している場合、サブスクリプションの数が多すぎると、アクティブなサブスクリプションを削除したときに、非アクティブであるが有効な設定済みサブスクリプションの1つが試行されます。定期的にトリガーされるサブスクリプション（デフォルトの最小値は100センチ秒）と、変更時にトリガーされるサブスクリプションがサポートされています。

サブスクリプションの設定では、NETCONF やその他のノースバウンド プログラマブル インターフェイス（RESTCONF、gNMI など）がサポートされています。

Cisco IOS XE システムのテレメトリでは、ダイナミック サブスクリプションと設定済みサブスクリプションの 2 種類のサブスクリプションが使用されます。

動的サブスクリプションは、パブリッシャに接続するクライアント（サブスクライバ）によって作成されるため、ダイヤルインと見なされます。設定済みサブスクリプションでは、パブリッシャは受信者への接続を開始し、その結果ダイヤルアウトと見なされます。

<establish-subscription> RPC の例

次に、<establish-subscription> RPC の例を示します。RPC では、stream、xpath-filter、および period の各フィールドが必須です。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<establish-subscription
xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
<stream>yp:yang-push</stream>
<yp:xpath-filter>/mdt-oper:mdt-oper-data/mdt-subscriptions</yp:xpath-filter>
<yp:period>1000</yp:period>
</establish-subscription>
</rpc>
```

YANG-push

YANG-push は、YANG データベース用のサブスクリプションおよびプッシュのメカニズムです。YANG-push サブスクリプションは、データ モデルを使用して定義されます。YANG-push を使用して、サブスクライバ アプリケーションは YANG データベースから更新の継続的なカスタマイズ ストリームを要求できます。YANG-push はデバイス上にインストールされている YANG モデルにより説明される構成データベース内と運用データベース内のすべてのデータを対象にしますすべてのデータのサブスクリプションがサポートされているわけではないので、データ用のフィルタを備える必要があります。

（注） YANG-push ストリームを指定する必要があります。

XPath フィルタのサポート

XMLパス言語（XPath）フィルタは、サブスクライブする情報要素を指定します。テレメトリパーサーに、必要なサブスクリプション情報がデータモデル内のどこにあるかを通知します。XPath フィルタの更新フィルタ グループがサブスクリプション用にサポートされます。

定期パブリケーション

定期サブスクリプションでは、サブスクライブ対象情報による最初のプッシュ更新は即時に送信されます。ただしデバイスがビジー状態であったりネットワークが混雑していたりすると遅延することがあります。次に更新は、設定された定期タイマーの満了時に送信されます。たとえば、期間を 10 分と設定すると、サブスクリプションの作成直後に最初の更新が送信され、その後は 10 分おきに送信されます。

期間とは、定期的なプッシュ更新間のセンチ秒（1/100 秒）単位の時間です。期間が 1000 であれば、サブスクライブ対象情報の更新は 10 秒ごとになります。最小の期間間隔は 100（つまり 1 秒）です。デフォルト値はありません。この値は <establish subscription> RPC に明示的に設定する必要があります。

現在存在しないデータのサブスクリプションは許可され、通常のサブスクリプションとして実行されます。空のデータをサブスクライブすると、要求された期間で空の更新通知が送信されます。

定期的な更新には、サブスクライブ対象のデータ要素またはテーブルのフルコピーが含まれます。

テレメトリの RPC サポート

テレメトリには <establish-subscription> RPC と <delete-subscription> RPC がサポートされています。

<establish-subscription> RPC が送信されると、パブリッシャからの RPC 応答には <rpc-reply> メッセージと、結果ストリングを含む <subscription-result> 要素が含まれます。

次の表は、<rpc-reply> メッセージでの応答と、応答の理由を示しています。

結果文字列	RPC	原因
ok	<establish-subscription> <delete-subscription>	成功
error-no-such-subscription	<delete-subscription>	指定されたテンプレートは存在しません。

結果文字列	RPC	原因
error-no-such-option	<establish-subscription>	要求されたサブスクリプションはサポートされていません。
error-insufficient-resources	<establish-subscription>	サブスクリプションは次の理由により作成できません。 • サブスクリプションが多すぎる。 • 要求されたデータの量が大きすぎると見なされる。 • 定期的なサブスクリプションの間隔が短すぎる。
error-other	<establish-subscription>	その他の何らかのエラーです。

テレメトリでの NETCONF セッション

テレメトリのサブスクリプションおよび更新は、NETCONF セッションを介して転送されます。テレメトリのサブスクリプションを確立するために使用されるNETCONFセッションは、テレメトリの更新を受け取ります。NETCONFセッションが破棄されたり、接続が切断された場合は、関連付けられたテレメトリ サブスクリプションも破棄されます。

すべてのセッションが NETCONF セッションであるため、すべてのセッションの制約は、NETCONF の実装に応じたものとなります。

テレメトリにおけるハイ アベイラビリティ

ダイナミック テレメトリの接続は、アクティブなスイッチもしくはスイッチ スタック内のメンバーへの SSH、またはハイ アベイラビリティ対応ルータにおけるアクティブなルート プロセッサへの SSH による NETCONF セッションで確立されます。切り替え後は、テレメトリのサブスクリプションを伝送する NETCONF セッションを含め、暗号を使用するすべてのセッションを破棄し、再確立する必要があります。切り替え後は、すべてのサブスクリプションも再作成する必要があります。

サンプルのモデル駆動型テレメトリ RPC

サブスクリプションの作成

サブスクリプションは、確立された NETCONF セッションを介して、XML RPC を使用して作成されます。<establish-subscription> RPC が IETF テレメトリのクライアントまたはコレクタからネットワーク デバイスに送信されます。RPC では、stream、xpath-filter、および period の各フィールドが必須です。

次に、運用データベースのサブスクリプション テーブルへのサブスクリプションの例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <establish-subscription
          xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
          xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <stream>yp:yang-push</stream>
        <yp:xpath-filter>/mdt-oper:mdt-oper-data/mdt-subscriptions</yp:xpath-filter>
        <yp:period>1000</yp:period>
      </establish-subscription>
    </rpc>
```

応答コードの受信

サブスクリプションが正常に作成されると、デバイスはサブスクリプション結果 notif-bis:ok およびサブスクリプション ID で応答します。次に、サンプル応答 RPC メッセージの例を示します。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
<subscription-result xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
xmlns:notif-bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:
ok</subscription-result>
<subscription-id
xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications">2147484201</subscription-id>
</rpc-reply>
```

サブスクリプションのプッシュ更新の受信

デバイスからプッシュされるサブスクリプション更新は XML RPC 形式であり、それらが作成された同じ NETCONF セッションにより送信されます。サブスクライブ対象情報の要素またはツリーは datastore-contents-xml タグ内で返されます。次に示すのは、サブスクライブ対象情報を提供するサンプル RPC メッセージです。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2017-05-09T21:34:51.74Z</eventTime>
    <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <subscription-id>2147483650</subscription-id>
        <datastore-contents-xml>
            <cpu-usage
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-process-cpu-oper"><cpu-utilization>
```

```
            <five-minutes>5</five-minutes></cpu-utilization></cpu-usage>
          </datastore-contents-xml>
    </push-update>
</notification>
```

サブスクリプションが行われる情報要素が空である場合、またはそれが動的（名前付きアクセス リストなど）であり存在しない場合、定期更新は空になり、自己終結 datastore-contents-xml タグを持つことになります。次に示すのは、定期更新が空であるサンプル RPC メッセージです。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2017-05-09T21:34:09.74Z</eventTime>
    <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <subscription-id>2147483649</subscription-id>
        <datastore-contents-xml />
    </push-update>
</notification>
```

サブスクリプションの詳細の取得

現在のサブスクリプションの一覧を取得するには、<get> RPC を Cisco-IOS-XE-mdt-oper モデルに送信します。現在のサブスクリプションの一覧を表示するには、**show telemetry ietf subscription** コマンドも使用できます。

次に、<get> RPC メッセージの例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
        <mdt-subscriptions/>
      </mdt-oper-data>
    </filter>
  </get>
</rpc>
```

次に、RPC 応答の例を示します。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>2147485164</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-xml</encoding>
          <period>100</period>
          <xpath>/ios:native/router/ios-rip:rip/ios-rip:version</xpath>
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <updates-in>0</updates-in>
```

```
        <updates-dampened>0</updates-dampened>
        <updates-dropped>0</updates-dropped>
      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
```

次に、**show telemetry ietf subscription dynamic brief** コマンドの出力例を示します。

```
Device# show telemetry ietf subscription dynamic brief

Telemetry subscription brief

  ID               Type        State       Filter type
  ---------------------------------------------------
  2147483667       Dynamic     Valid       xpath
  2147483668       Dynamic     Valid       xpath
  2147483669       Dynamic     Valid       xpath
```

次に、**show telemetry ietf subscription** *subscription-ID***detail** コマンドの出力例を示します。

```
Device# show telemetry ietf subscription 2147483667 detail

Telemetry subscription detail:

  Subscription ID: 2147483667
  State: Valid
  Stream: yang-push
  Encoding: encode-xml
  Filter:
    Filter type: xpath
    XPath: /mdt-oper:mdt-oper-data/mdt-subscriptions
  Update policy:
    Update Trigger: periodic
    Period: 1000
  Notes:
```

サブスクリプションの削除

テレメトリ サブスクリプションは、2 つの方法で削除できます。1 つ目は、subscription-id タグにサブスクリプション ID を含む <delete-subscription> RPC を送信することです。これは、サブスクライバのみが実行できます。サブスクリプションは、親 NETCONF セッションが破棄または切断されたときにも削除されます。ネットワーク接続が中断された場合は、SSH/NETCONF セッションがタイムアウトしてその後にサブスクリプションが削除されるまで、多少の時間がかかることがあります。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <delete-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
        <subscription-id>2147483650</subscription-id>
    </delete-subscription>
</rpc>
```

モデル駆動型テレメトリに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
NETCONF-YANG パッチ	https://tools.ietf.org/wg/netconf/draft-ietf-netconf-yang-patch/
YANG エクスプローラ	https://github.com/CiscoDevNet/yang-explorer

標準および RFC

標準/RFC	タイトル
イベント通知の *NETCONF* サポート	draft-ietf-netconf-netconf-event-notifications-01
RFC 6241	ネットワーク設定プロトコル（NETCONF）
イベント通知への登録	draft-ietf-netconf-rfc5277bis-01
YANG データストア プッシュのサブスクライブ	draft-ietf-netconf-yang-push-04

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

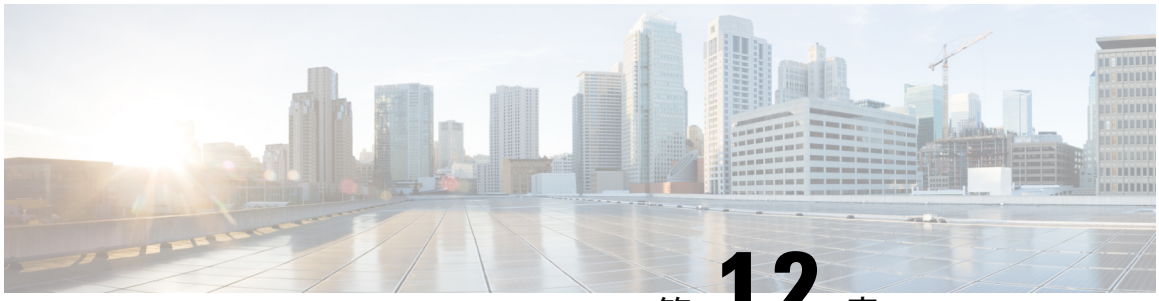
モデル駆動型テレメトリの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *18* : モデル駆動型テレメトリの機能情報

機能名	リリース	機能情報
モデル駆動型テレメトリ	Cisco IOS XE Everest 16.6.1	モデル駆動型テレメトリでは、ネットワーク デバイスからサブスクライバに、リアルタイムの設定や運用状態の情報を継続的にストリームすることができます。 この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。



第 12 章

In Service Model Update

このモジュールでは、In Service Model Update によりデバイス上の YANG データ モデルを更新する方法を説明します。

このモジュールの構成は次のとおりです。

In Service Model Update について

In Service Model Update の概要

In Service Model Update は、新しいデータ モデルまたは拡張機能を既存のデータ モデルに追加します。In Service Model Update は、リリース サイクル外の YANG モデル拡張機能を提供します。更新プログラム パッケージはすべての既存のモデルの上位セットです。これには、更新された YANG モデルを始めとするすべての既存モデルが含まれています。

データ モデル インフラストラクチャは、Cisco IOS XE デバイス用の YANG モデル定義管理インターフェイスを実装します。データ モデル インフラストラクチャは、Cisco IOS XE デバイスからノースバウンドに NETCONF インターフェイスを公開します。サポートされているデータ モデルには、IETF などの業界標準モデルと、Cisco IOS XE デバイス固有のモデルが含まれます。

In Service Model Update によって提供される機能は、その後の Cisco IOS XE ソフトウェア メンテナンス リリースに統合されます。データ モデル更新プログラム パッケージは、シスコ ソフトウェア ダウンロード センター からダウンロードできます。

In Service Model Update パッケージの互換性

更新プログラム パッケージは、イメージごとに構築されています。

更新プログラム パッケージのすべてのコンテンツは、将来のメインライン リリースまたはメンテナンス リリースのイメージの一部になります。イメージとプラットフォームのバージョンは、パッケージの追加およびアクティブ化の際に、In Service Model Update コマンドによってチェックされます。イメージまたはプラットフォームの不一致が発生すると、パッケージのインストールが失敗します。

更新プログラム パッケージの命名規則

In Service Model Update は、.bin ファイルとしてパッケージ化されています。このファイルには、特定のリリースおよびプラットフォームのすべての更新プログラムと、Readme ファイルが含まれています。これらのファイルにはリリース日があり、追加モデルの更新をともなって定期的に更新されます。

データ モデルの更新プログラム パッケージの命名規則は、次の形式に従っています。プラットフォームの種類-ライセンス レベル.リリース バージョン.DDTS ID-ファイル。次に、データ モデル更新ファイルの例を示します。

- isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

Readme ファイルは、次の情報を提供します。

- データ モデルのアクティブ化または非アクティブ化中に表示されるコンソール メッセージおよびエラー メッセージ
- データ モデルのインストールによる影響
- 副作用と考えられる回避策
- In Service Model Update によって影響を受けるパッケージ
- リスタートのタイプ

更新プログラム パッケージのインストール

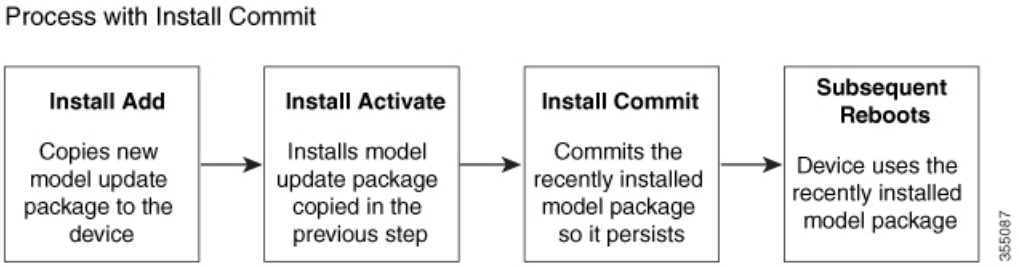
デバイスに In-Service Model Update パッケージをインストールするには、**install add**、 **install activate**、および **install commit** コマンドを特権 EXEC モードで使用します。

install add コマンドは、更新パッケージをリモートの場所からデバイスにコピーします。パッケージをコピーするには他の方法も使用できますが、その場合も、インストールしたプログラムを動作させるために **install add** コマンドを有効化する必要があります。 **install activate** コマンドを動作させるには、パッケージをデバイスのブートフラッシュで使用可能にする必要があります。**install commit** コマンドを有効化して、更新プログラムをリロード全体にわたって確定します。

更新プログラムをインストールすると、以前にインストールされたデータ モデルがある場合、それは置き換えられます。デバイスには常に、1 つの更新プログラムのみがインストールされます。データ モデル パッケージには、すべての更新された YANG モデルと、以前にデバイスにインストールされたすべての既存 YANG モデルが含まれています。

次のフロー チャートでは、モデル更新プログラム パッケージの動作を説明します。

図 *2*：モデル更新プログラム パッケージのコミット



パッケージをアクティブ化する際にNETCONF-YANGが有効化されていると、NETCONFプロセスがリスタートされます。すべてのアクティブなNETCONFセッションは、パッケージのアクティブ化中に破棄されます。パッケージの検証中にエラーが発生すると、アクティブ化プロセスは終了します。

更新プログラム パッケージの非アクティブ化

更新パッケージを非アクティブ化するには、**install deactivate** コマンドを使用します。変更を確定するには、**install commit** コマンドを有効化します。

表 *19*：モデル更新プログラム パッケージの非アクティブ化

操作	使用コマンド
パッケージの削除	**install remove** コマンドを使用します。 （注） パッケージを削除する前に非アクティブ化します。
パッケージの非アクティブ化	**install deactivate** コマンドを使用し、その後に **install commit** コマンドを使用します。 （注） **install commit** コマンドの使用が必要なのは、モデルパッケージの非アクティブ化をリロード全体にわたって確定するためです。非アクティブ化がコミットされていないと、その後にパッケージを削除しようとしても失敗します。

更新プログラムを非アクティブ化する際に、2つ以上のモデル更新プログラム パッケージがインストールされている場合、最も最近コミットされたモデル更新プログラム パッケージがデバイスによって使用されるモデル パッケージになります。以前にコミットされたその他のモデル パッケージがない場合、標準的なイメージとともに含まれているベース バージョンのデータ モデルが使用されるようになります。

更新プログラム パッケージのロールバック

ロールバックは、デバイスを更新前の動作状態に戻すメカニズムを提供します。ロールバック後は、変更が表示されるようになる前に NETCONF-YANG プロセスが再始動します。

更新は、**install rollback** コマンドを使用して、基本バージョン、最終コミットバージョン、または既知のコミット ID までロールバックできます。

In Service Model Update の管理方法

更新プログラム パッケージの管理

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**install add file tftp:** *filename* 例： `Device# install add file tftp://172.16.0.1//tftpboot/folder1/ isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`	リモートロケーションから（FTP、TFTP 経由で）デバイスにモデル更新プログラム パッケージをコピーし、プラットフォームとイメージのバージョンの互換性チェックを実行します。 • 他の方法を使用してリモートの場所からデバイスに更新パッケージをコピーすることもできます。ただし、その場合もパッケージをアクティブにする前に **install add** コマンドを実行する必要があります。
ステップ 3	**install activate file bootflash:** *filename* 例： `Device# install activate file bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`	更新パッケージが **install add** コマンドにより追加されていることを確認し、NETCONF プロセスを再開します。 • 更新パッケージをアクティブにする前に **install add** 操作を実行します。
ステップ 4	**install commit** 例：	リロードが繰り返されても持続する変更を行います。

	コマンドまたはアクション	目的
	`Device# install commit`	• NETCONF プロセスは再開されません。
ステップ 5	**install deactivate file bootflash:** *filename* 例： `Device# install deactivate file bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`	指定された更新プログラム パッケージを非アクティブにして、NETCONF プロセスを再開します。
ステップ 6	**install commit** 例： `Device# install commit`	リロードが繰り返されても持続する変更を行います。 • NETCONF プロセスは再開されません。
ステップ 7	**install rollback to** {**base** | **committed** | **id** *commit-ID*} 例： `Device# install rollback to base`	更新を基本バージョン、最後にコミットしたバージョン、または既知のコミット ID にロールバックし、NETCONF プロセスを再起動します。 • *commit-id* 引数の有効な値は 1 ～ 4294967295 です。 • データ モデル更新の古いバージョンが使用可能です。
ステップ 8	**install remove** {**file bootflash:** *filename* | **inactive**} 例： `Device# install remove file bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`	指定された更新プログラム パッケージをブートフラッシュから削除します。 • パッケージは削除する前に非アクティブにする必要があります。
ステップ 9	**show install summary** 例： `Device# show install summary`	アクティブ パッケージに関する情報を表示します。 • このコマンドの出力は、設定されている **install** コマンドに応じて変化します。

In Service Model Update の構成例

例：更新プログラム パッケージの管理

次の例で使用しているのは、Cisco 4000 シリーズ サービス統合型ルータのサンプル イメージです。

次の例では、モデル更新プログラム パッケージ ファイルの追加方法を示しています。

```
Device# install add file tftp://172.16.0.1//tftpboot/folder1/
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

install_add: START Sun Feb 26 05:57:04 UTC 2017
Downloading file
tftp://172.16.0.1//tftpboot/folder1/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Finished downloading file
tftp://172.16.0.1//tftpboot/folder1/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
to bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
SUCCESS: install_add /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Sun Feb 26 05:57:22 UTC 2017
Device#
```

次に、更新パッケージ ファイルをデバイスに追加した後の **show install summary** コマンドの出力例を示します。

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Committed Packages:
No packages
Uncommitted Packages:
No packages
Device#
```

次の例では、追加された更新プログラム パッケージ ファイルをアクティブにする方法を示しています。

```
Device# install activate file bootflash:
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

install_activate: START Sun Feb 26 05:58:41 UTC 2017
DMP package.
Netconf processes stopped
SUCCESS: install_activate /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Sun Feb 26 05:58:58 UTC 2017*Feb 26 05:58:47.655: %DMI-4-CONTROL_SOCKET_CLOSED:
SIP0: nesd: Confd control socket closed Lost connection to ConfD (45): EOF on socket to
 ConfD.
*Feb 26 05:58:47.661: %DMI-4-SUB_READ_FAIL: SIP0: vtyserverutild:
Confd subscription socket read failed Lost connection to ConfD (45):
EOF on socket to ConfD.
*Feb 26 05:58:47.667: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: syncfd:
Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 05:59:43.269: %DMI-5-SYNC_START: SIP0: syncfd:
```

```
External change to running configuration detected.
The running configuration will be synchronized to the NETCONF running data store.
*Feb 26 05:59:44.624: %DMI-5-SYNC_COMPLETE: SIP0: syncfd:
The running configuration has been synchronized to the NETCONF running data store.
Device#
```

次に示すのは、**show install summary** コマンドがモデル パッケージのステータスをアクティブでありコミット未完了と表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
No packages
Uncommitted Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device#
```

次の例では、**install commit** コマンドの実行方法を示しています。

```
Device# install commit

install_commit: START Sun Feb 26 06:46:48 UTC 2017
SUCCESS: install_commit Sun Feb 26 06:46:52 UTC 2017
Device#
```

次に示すのは、**show install summary** コマンドが、更新パッケージがコミットされてリロードが繰り返されても持続することを表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Uncommitted Packages:
No packages
Device#
```

次の例は、更新プログラム パッケージを基本パッケージにロールバックする方法を示しています。

```
Device# install rollback to base

install_rollback: START Sun Feb 26 06:50:29 UTC 2017
7 install_rollback: Restarting impacted processes to take effect
7 install_rollback: restarting confd
*Feb 26 06:50:34.957: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: syncfd:
Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 06:50:34.962: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: nesd:
Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 06:50:34.963: %DMI-4-SUB_READ_FAIL: SIP0: vtyserverutild:
Confd subscription socket read failed Lost connection to ConfD (45):
EOF on socket to ConfD.Netconf processes stopped
7 install_rollback: DMP activate complete
```

```
SUCCESS: install_rollback Sun Feb 26 06:50:41 UTC 2017
*Feb 26 06:51:28.901: %DMI-5-SYNC_START: SIP0: syncfd:
External change to running configuration detected.
The running configuration will be synchronized to the NETCONF running data store.
*Feb 26 06:51:30.339: %DMI-5-SYNC_COMPLETE: SIP0: syncfd:
The running configuration has been synchronized to the NETCONF running data store.
Device#
```

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

```
Device# show install package bootflash:
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

Name: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Version: 16.5.1.0.199.1484082952..Everest
Platform: ISR4300
Package Type: dmp
Defect ID: CSCxxxxxxx
Package State: Added
Supersedes List: {}
Smu ID: 1
Device#
```

次の NETCONF hello メッセージの例では、新規データ モデル パッケージのバージョンを確認します。

```
Getting Capabilities: (admin @ 172.16.0.1:830)
PROTOCOL netconf
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
<capability>urn:ietf:params:netconf:capability:notification:1.0</capability>
<capability>urn:ietf:params:netconf:capability:interleave:1.0</capability>
<capability>http://tail-f.com/ns/netconf/actions/1.0</capability>
<capability>http://tail-f.com/ns/netconf/extensions</capability>
<capability>urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=
explicit&also-supported=report-all-tagged</capability>
<capability>urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults?
revision=2011-06-01&module=ietf-netconf-with-defaults</capability>
<capability>http://cisco.com/ns/yang/Cisco-IOS-XE-aaa?module=
Cisco-IOS-XE-aaa&revision=2017-02-07</capability>
<<capability>http://cisco.com/ns/yang/Cisco-IOS-XE-native?module=
Cisco-IOS-XE-native&revision=2017-01-07&features=virtual-
template,punt-num,multilink,eth-evc,esmc,efp,dot1x</capability>
Device#
```

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

```
Device# show install log

[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add( FATAL)]: File path (scp) is not yet supported for this command
```

```
[4|install_add]: START Sun Feb 26 05:57:04 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

Sun Feb 26 05:57:22 UTC 2017
[5|install_activate]: START Sun Feb 26 05:58:41 UTC 2017
Device#
```

次の例で使用しているのは、Cisco Catalyst 3000 シリーズ スイッチのサンプル イメージです。

次の例では、モデル更新プログラム パッケージ ファイルの追加方法を示しています。

```
Device# install add file tftp://172.16.0.1//tftpboot/folder1/
cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin

install_add: START Sat Jul 29 05:57:04 UTC 2017
Downloading file tftp://172.16.0.1//tftpboot/folder1/
cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Finished downloading file tftp://172.16.0.1//tftpboot/folder1/
cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.SPA.smu.bin
to bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
SUCCESS: install_add /bootflash/cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Sat Jul 29 05:57:22 UTC 2017
Device#
```

次に示すのは、**show install summary** コマンドが、更新パッケージがコミットされてリロードが繰り返されても持続することを表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Uncommitted Packages:
No packages
Device#
```

In Service Model Update の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *20 : In Service Model Update* の機能情報

機能名	リリース	機能情報
In Service Model Update	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	このモジュールでは、In Service Model Update で YANG データ モデルを更新する方法を説明します。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco クラウド サービス ルータ 1000v • Cisco サービス統合型仮想ルータ（ISRv） コマンド **install (Programmability)**、**show install (Programmability)** が導入または更新されました。
	Cisco IOS XE Everest 16.6.1	Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ