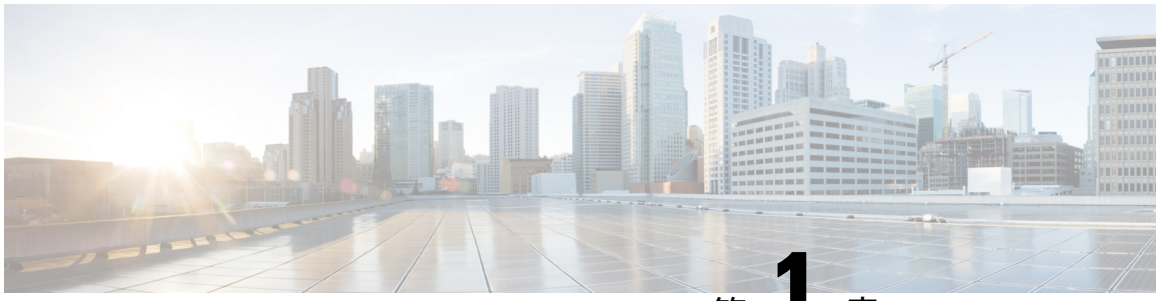
Cisco NCS 560 シリーズ ルータ（IOS XR リリース 7.0.x）システム セットアップおよびソフトウェア インストール ガイド

初版：2019 年 8 月 30 日

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

目 次



第 1 章

Cisco NCS 560-4 製品の概要

Cisco NCS 560-4 ルータは、完全な冗長性を備えた 4 ラック ユニット（4-RU）の中央集中型フォワーディング システムで、次の機能があります。

- ルータ プロセッサ（RSP）スロット（2 個）
- インターフェイス モジュール（IM）スロット（6 個）
- 合計 1.8 Tbps のバックプレーン容量、すべての IM スロットで 25 Gbps 対応の SerDes を装着

  シャーシに約 1.5 KW 電力を供給可能な（2 + 1）電源モジュールのサポート

  ルート プロセッサの設定に基づいた拡張温度のサポート

Cisco NCS 560-4 ルータの詳細については、『*Cisco NCS 560-4 Router Hardware Installation Guide*』を参照してください。

Cisco NCS 560-4 ルータは、次のルート プロセッサをサポートしています。

- N560-RSP4：中規模ルート プロセッサ
- N560-RSP4-E：合計 800 Gbps のスイッチング容量を備えた高性能ルータ プロセッサ。

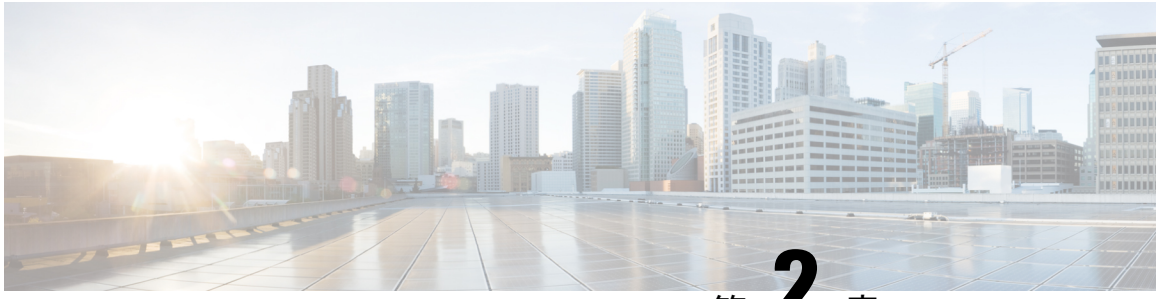
（注） 上記のルート プロセッサは同じルータで一緒に使用することはできません。

詳細については、『*Cisco N560-RSP4 and Cisco N560-RSP4-E Route Processor Hardware Installation Guide*』を参照してください。

- コマンド モード （2 ページ）

コマンド モード

コマンド モード	説明
XR EXEC モード （XR VM 実行モード）	XR VM でコマンドを実行してルータの動作状態を表示します。 例： `RP/0/RP0/cpu 0: routerRP0/CPU0:ios#`
XR コンフィギュレーション モード （XR VM コンフィギュレーション モード）	XR VM でセキュリティやルーティングなどの XR 機能を設定します。 例： `RP/0/RP0/cpu 0: routerRP0/CPU0:ios#`**`configure`** `RP/0/RP0/cpu 0: router(config)#`
システム管理 EXEC モード （システム管理 実行モード）	システム管理 でコマンドを実行して、ルータ ハードウェアの動作状態を表示およびモニタします。シャーシまたは個別のハードウェア モジュールは、このモードでリロードすることができます。 例： `RP/0/RP0/cpu 0: router#`**`admin`** `sysadmin-vm:0_RP0#`
システム管理コンフィギュレーション モード （システム管理コンフィギュレーション モード）	システム管理 VM でコンフィギュレーション コマンドを実行して、シャーシ全体のハードウェア モジュールを管理および操作します。 例： `RP/0/RP0/cpu 0: routerRP0/CPU0:ios#`**`admin`** `sysadmin-vm:0_RP0#`**`config`** `sysadmin-vm:0_RP0(config)#`



第 2 章

ルータの起動

ハードウェアの設置後、ルータを起動します。XR コンソール ポートに接続し、ルータの電源をオンにします。ルータは、プリインストールされたオペレーティング システム（OS）イメージを使用してブート プロセスを実行します。ルータ内に使用できるイメージがない場合は、iPXE ブートまたは外部のブート可能な USB ドライブを使用してルータを起動できます。

起動が完了したら、root ユーザ名とパスワードを作成します。その組み合わせを使って XR コンソールにログインするとルータ プロンプトが表示されます。XR コンソールで作成された最初のユーザは、システム管理コンソールに同期されます。XR コンソールからシステム管理コンソールにアクセスして、システム管理設定を行います。

ルータの起動

新しいルータに接続するには、ルート プロセッサ（RP）のコンソール ポートを使用します。コンソール ポートはデフォルトでXR コンソールに接続されます。必要に応じて、設定済みの管理ポートを通じてさらに接続を確立できます。

手順

ステップ 1 RP のコンソール ポートに端末を接続します。

ステップ 2 ワークステーションで端末エミュレーション プログラムを起動します。

モジュラ型シャーシ RP の場合、コンソール設定はボー レートが 9600 bps、パリティなし、ストップ ビットが 2、データ ビットが 8 です。固定シャーシの場合、コンソール設定はボー レートが 115200 bps、パリティなし、ストップ ビットが 2、データ ビットが 8 です。ボー レートはデフォルトで設定されていて、変更することはできません。

ステップ 3 ルータの電源を投入します。

電源コードを電源モジュールに接続してルータを起動します。端末エミュレーションプログラムのコンソール画面に、ブート プロセスの詳細が表示されます。

ステップ 4 Enter を押します。

root-system ユーザ名の入力を求めるプロンプトが表示されたらブート プロセスは完了です。プロンプトが表示されない場合は、ルータの初期ブート手順が完了するまでしばらく待ってから Enter を押してください。

重要 ブート プロセスが失敗する原因として、ルータにプリインストールされているイメージが破損していることが考えられます。この場合は、外部のブート可能な USB ドライブを使用してルータを起動できます。

次のタスク

root ユーザ名およびパスワードを指定します。

root ユーザ クレデンシャルの設定

ルータの初回起動時に、root クレデンシャル（ユーザ名とパスワード）の設定を求めるプロンプトが表示されます。これらは、XR（root-lr）コンソールおよびシステム管理 VM（root-system）の root ユーザ クレデンシャル、およびディザスタ リカバリのクレデンシャルとして設定されます。

始める前に

ブート プロセスを完了する必要があります。ブート プロセスの開始方法については、ルータの起動 （3 ページ） を参照してください。

手順

ステップ 1 **Enter root-system username:** *username*

root ユーザのユーザ名を入力します。文字数制限は 1023 文字です。この例では、root ユーザの名前は「root」です。

重要 指定したユーザ名は、XR コンソールの「root-lr」グループにマッピングされます。また、システム管理コンソールの「root-system」ユーザとしてもマッピングされます。

ルータの初回起動時またはイメージの再作成後は、ルータにユーザ設定がありません。この場合、ルータによって「root-system ユーザ名」を指定するように要求されます。ただしすでにルータが設定されている場合は、ステップ 4 で説明したように「ユーザ名」の入力を求めるプロンプトが表示されます。

ステップ 2 **Enter secret:** *password*

root ユーザのパスワードを入力します。パスワードの文字数は 6 ～ 253 文字です。セキュリティ上の理由から、入力したパスワードは CLI に表示されません。

root ユーザにはスーパーユーザ権限があるため、root ユーザ名とパスワードは保護する必要があります。これはルータ設定全体へのアクセスに使用されます。

ステップ 3 **Enter secret again:** *password*

root ユーザのパスワードをもう一度入力します。パスワードは、前のステップで入力したパスワードと一致しないと拒否されます。セキュリティ上の理由から、入力したパスワードは CLI に表示されません。

ステップ 4 **Username:** *username*

XR VM コンソールにログインするため、root-system ユーザ名を入力します。

ステップ 5 **Password:** *password*

root ユーザのパスワードを入力します。正しいパスワードを入力するとルータのプロンプトが表示されます。これで XR VM コンソールにログインできました。

ステップ 6 （任意） **show run username**

ユーザの詳細を表示します。

```
username root
 group root-lr
 group cisco-support
 secret 5 $1$NBg7$fHs1inKPZVvzqxMv775UE/
!
```

次のタスク

- XR コンソールからルーティング機能を設定します。
- システム管理プロンプトでシステム管理設定を行います。システム管理プロンプトは、システム管理コンソールへのアクセス時に表示されます。システム管理プロンプトを表示する方法については、システム管理コンソールへのアクセス （10 ページ）を参照してください。

コミット チェックの検証

commit 操作には、システム内のさまざまなコンポーネントに対して 1 つまたは複数の設定が含まれています。ただし、個々の設定が相互に依存し、競合する場合は、commit 操作がすべてのコンポーネントで正常に実行されない可能性があります。

設定をコミットする前に、**validate commit** 操作を使用して、動作状態または利用可能なハードウェア リソースを参照せずに全体的な設定を検証します。コマンドは、内部中央集中型設定検証（CCV）を使用して設定を検証します。設定が失敗した場合は、競合の重大度に基づいて警告またはエラー メッセージが表示されます。

手順

ステップ 1 **configuration validation enable**

例：

```
RP/0/RP0/cpu 0: router(config)#configuration validation enable
```

コミット チェック操作を有効にします。

ステップ 2 **configuration validation failure-type unsupported report**

例：

```
RP/0/RP0/cpu 0: router(config)#configuration validation failure-type unsupported report
```

コミット チェック操作を有効にします。

ステップ 3 **show configuration validation internal ccv status**

例：

```
RP/0/RP0/cpu 0: router#show configuration validation internal ccv status
Current status of CCV:
CCV enablement state: Enabled
        user-requested:   True
        dirty:            False
        unstable:         False
        repopulate:       Not required
        init-sync:        True
    CCV unsupported:      Enabled
    YVE database dump:    Disabled
        Remaining dumps:  0
        Dump to:
        Restricted to:
        Dump original:    False
        Dump final:       False
        Data types:       NONE
        Include failures: False
    YVE debug terminal:
```

コミット チェック操作が有効になっていることを確認します。

ステップ 4 **show configuration validation internal shadow**

例：

```
RP/0/RP0/cpu 0: router#show configuration validation internal shadow
```

現在のシャドウ設定を確認します。このコマンドは BGP と RPL の設定に固有のものです。

ステップ 5 **show configuration validation unsupported**

例：

```
RP/0/RP0/cpu 0: router#show configuration validation unsupported
Aug  1 23:53:12.590 IST
SEMANTIC ERRORS: This configuration was rejected by the system due to semantic errors.
The individual errors with each failed configuration command can be found below.
hostname
UNSUPPORTED: Configuration validation is not supported on this item group ccv
UNSUPPORTED: Configuration validation is not supported on this item end-group group key

UNSUPPORTED: Configuration validation is not supported on this item end-group group tcp

UNSUPPORTED: Configuration validation is not supported on this item end-group group RSVP

UNSUPPORTED: Configuration validation is not supported on this item
```

サポートされていない CLI を確認します。

ステップ 6 **validate commit**

例：

```
RP/0/RP0/cpu 0: router(config)#validate commit
```

設定を検証します。コマンドを実行すると、設定の結果が表示されます。エラー メッセージが表示された場合は、エラーを修正します。

ステップ 7 **commit**

例：

```
RP/0/RP0/cpu 0: router(config)#commit
```

設定をコミットします。

例

例：**RPL-BGP 設定のコミット確認**

次に、この設定がコミットされた場合の RPL と BGP の設定例とシャドウ検証の出力例を示します。

RPL 設定：ルート ポリシーの設定:

```
route-policy IBGP-TEST-out
  set next-hop self
end-policy
!
route-policy IBGP-TEST-v6-out
  set next-hop self
end-policy
!
route-policy EBGP-FREE-PEERS-out
  set next-hop self
end-policy
!
route-policy EBGP-FREE-PEERSv6-out
  set next-hop self
```

```
end-policy
!
```

BGP 設定：

```
router bgp 32934
 nsr
 bgp router-id 172.16.0.0
 bgp graceful-restart
 address-family ipv4 unicast
  maximum-paths ebgp 32
  maximum-paths ibgp 32
 !
 address-family ipv6 unicast
 !
 address-family ipv6 multicast
 !
 neighbor-group <name>
  remote-as 325
  update-source FortyGigE0/7/0/20
  address-family ipv4 unicast
   multipath
   maximum-prefix 700000 75
   soft-reconfiguration inbound
  !
 !
 neighbor-group <name1>
  remote-as 32934
  update-source FortyGigE0/7/0/20
  address-family ipv6 unicast
   multipath
   route-policy IBGP-TEST-v6-out out
   soft-reconfiguration inbound
  !
 !
 neighbor-group FREE_PEERS
  address-family ipv4 unicast
   maximum-prefix 2 80 restart 240
   route-policy EBGP-FREE-PEERS-out out
   remove-private-AS
  !
 !
 neighbor-group IBGP-TEST
  remote-as 32934
  update-source Loopback0
  address-family ipv4 unicast
   multipath
   maximum-prefix 700000 75
   route-policy IBGP-TEST-out out
   soft-reconfiguration inbound
  !
 !
 neighbor-group FREE_PEERSv6
  remote-as 100
  address-family ipv4 unicast
   route-policy EBGP-FREE-PEERS-out out
  !
  address-family ipv6 unicast
   maximum-prefix 2 80 restart 240
   route-policy EBGP-FREE-PEERSv6-out out
   remove-private-AS
  !
 !
```

この RPL と BGP の設定の場合は、commit check を有効にすると、シャドウ設定に次の出力が表示されます。

```
RP/0/RP0/CPU0:Router#show configuration validation internal shadow
Wed Jul 11 21:58:16.139 UTC
Building configuration...
!! IOS XR Configuration version = <version>
!! Last configuration change at Wed Jul 11 21:36:44 2018 by root
!
route-policy IBGP-TEST-out
  set next-hop self
end-policy
!
route-policy IBGP-TEST-v6-out
  set next-hop self
end-policy
!
route-policy EBGP-FREE-PEERS-out
  set next-hop self
end-policy
!
route-policy EBGP-FREE-PEERSv6-out
  set next-hop self
end-policy
!
router bgp 32934
 nsr
 bgp router-id 172.16.0.0
 bgp graceful-restart
 address-family ipv4 unicast
  maximum-paths ebgp 32
  maximum-paths ibgp 32
 !
 address-family ipv6 unicast
 !
 address-family ipv6 multicast
 !
 neighbor-group <name>
  remote-as 325
  update-source FortyGigE0/7/0/20
  address-family ipv4 unicast
   multipath
   maximum-prefix 700000 75
   soft-reconfiguration inbound
  !
 !
 neighbor-group <name1>
  remote-as 32934
  update-source FortyGigE0/7/0/20
  address-family ipv6 unicast
   multipath
   route-policy IBGP-TEST-v6-out out
   soft-reconfiguration inbound
  !
 !
 neighbor-group FREE_PEERS
  address-family ipv4 unicast
   maximum-prefix 2 80 restart 240
   route-policy EBGP-FREE-PEERS-out out
   remove-private-AS
  !
 !
 neighbor-group IBGP-TEST
  remote-as 32934
```

```
  update-source Loopback0
  address-family ipv4 unicast
   multipath
   maximum-prefix 700000 75
   route-policy IBGP-TEST-out out
   soft-reconfiguration inbound
  !
 !
 neighbor-group FREE_PEERSv6
  remote-as 100
  address-family ipv4 unicast
   route-policy EBGP-FREE-PEERS-out out
  !
  address-family ipv6 unicast
   maximum-prefix 2 80 restart 240
   route-policy EBGP-FREE-PEERSv6-out out
   remove-private-AS
  !
!
```

例：コミット チェック検証の失敗

```
Router(config)#int Bundle-ether 1
Router(config-if)#dampening 1 2 2 4
Router(config-if)#validate commit show
Mon Jul 30 13:04:18.433 IST

Validation of configuration items failed during validate commit.
SEMANTIC ERRORS: This configuration was rejected by the system due to semantic errors.
The individual errors with each failed configuration command can be found below.
interface Bundle-Ether1
 dampening 1 2 2 4
ERROR: Suppress threshold on "Bundle-Ether1" should be greater than reuse threshold '2',
 but value provided is '2'
end

Router(config-if)#commit show-error
Mon Jul 30 13:04:32.958 IST
Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted.
SEMANTIC ERRORS: This configuration was rejected by the system due to semantic errors.
The individual errors with each failed configuration command can be
found below.

interface Bundle-Ether1
 dampening 1 2 2 4
This operation is not supported: Reuse threshold is not less than supressed threshold
end
```

システム管理コンソールへのアクセス

すべてのシステム管理とハードウェア管理の設定を行うには、XR コンソールからシステム管理コンソールにログインする必要があります。

手順

ステップ 1 root ユーザとして XR コンソールにログインします。

ステップ 2 **admin**

例：

ログイン バナーは、デフォルトで有効に設定されています。次の例では、ログイン バナーを有効にした状態のコマンド出力を示しています。

```
RP/0/RP0/cpu 0: routerRP0/CPU0:ios#admin

Mon May 22 06:57:29.350 UTC

root connected from 127.0.0.1 using console on host
sysadmin-vm:0_RP0# exit
Mon May  22 06:57:32.360 UTC
```

次の例では、ログイン バナーを無効にした状態のコマンド出力を示しています。

```
RP/0/RP0/CPU0:router#admin
Thu Mar 01:07:14.509 UTC
sysadmin-vm:0_RP0# exit
```

ステップ 3 （任意） **exit**

システム管理モードから XR モードに戻ります。

管理ポートの設定

管理ポートをシステム管理およびリモート通信に使用するには、管理イーサネット インターフェイスの IP アドレスとサブネット マスクを設定する必要があります。他のネットワーク上のデバイス（リモート管理ステーションや TFTP サーバなど）と通信する場合は、ルータのデフォルト（スタティック）ルートを設定する必要があります。

始める前に

- ネットワーク管理者またはシステムの設計担当者に問い合わせて、管理インターフェイスの IP アドレスおよびサブネット マスクを入手します。
- RP の物理ポート イーサネット 0 は管理ポートです。ポートが管理ネットワークに接続されていることを確認します。

手順

ステップ 1 **configure**

ステップ 2 **interface MgmtEth** *rack/slot/port*

例：

```
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
```

プライマリ RP の管理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。

ステップ 3 **ipv4 address** *ipv4-address subnet-mask*

例：

```
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.1.1.1/8
```

IP アドレスとサブネット マスクをインターフェイスに割り当てます。

ステップ 4 **no shutdown**

例：

```
RP/0/RP0/CPU0:ios(config-if)#no shutdown
```

インターフェイスを「アップ」状態にします。

ステップ 5 **exit**

例：

```
RP/0/RP0/CPU0:ios(config-if)#exit
```

管理インターフェイス コンフィギュレーション モードを終了します。

冗長ルート プロセッサで上記の手順を繰り返します。

ステップ 6 **ipv4 virtual address** *ipv4 virtual address subnet-mask*

例：

```
RP/0/RP0/CPU0:ios(config)#ipv4 virtual address 1.70.31.160 255.255.0.0
```

仮想 IP アドレスとサブネット マスクをインターフェイスに割り当てます。

ステップ 7 **commit**

次のタスク

管理ポート経由でイーサネット ネットワークに接続します。端末エミュレーション プログラムで、管理インターフェイス ポートへの SSH または Telnet 接続をその IP アドレスを使って確立します。ルータに対して許可される Telnet セッションの数を設定するには、Telnet セッションを確立する前に、XR コンフィギュレーション モードで **telnet ipv4|ipv6 server max-servers** コマンドを使用します。

NTP サーバとのクロック同期の実行

XR コンソールとシステム管理コンソールにはそれぞれのシステム クロックがあります。これらのクロックが実際の時間とずれないように、NTP サーバのクロックと同期する必要があります。このタスクでは、XR コンソール用に NTP サーバを設定します。XR コンソールのクロックを同期すると、システム管理コンソールのクロックは自動的に XR コンソールのクロックと同期されます。

始める前に

管理ポートを設定して接続します。

手順

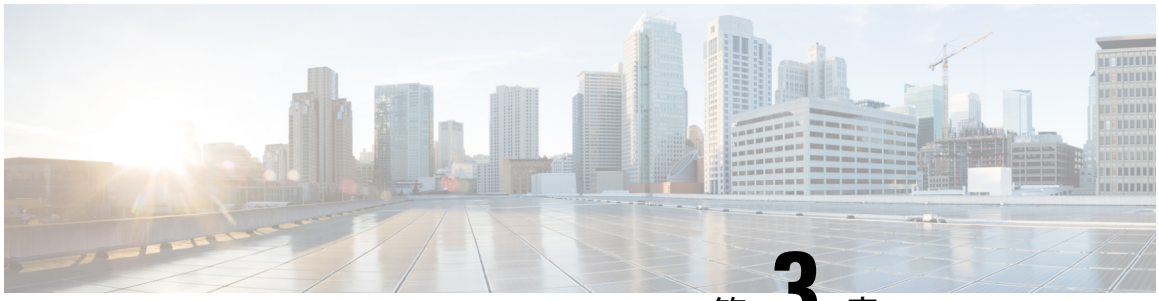
ステップ 1 **configure**

ステップ 2 **ntp server** *server_address*

例：

```
RP/0/RP0/cpu 0: routerRP0/CPU0:ios(config)#ntp server 64.90.182.55
```

指定したサーバと同期するように XR コンソールのクロックが設定されます。



第 3 章

予備チェックの実行

コンソールに正常にログインしたら、予備チェックを実行してデフォルト設定を確認する必要があります。チェックの実行時に設定の問題が検出された場合は、さらに設定を行う前に修正を行ってください。予備チェックの内容は次のとおりです。

ハードウェア モジュールのステータスの確認

ハードウェア モジュールには RP、ファン トレイなどがあります。ルータには複数のハードウェア モジュールが取り付けられています。すべてのハードウェア モジュールが正しく取り付けられて動作していることを確認するには、次のタスクを実行します。

始める前に

必要なハードウェア モジュールがすべてルータに取り付けられていることを確認します。

ノード ステータスの確認

ルータ上の各カードはノードを表します。ノードの動作ステータスは、**show platform** コマンドを使用して確認します。このコマンドは、XR およびシステム管理モードの両方の CLI で個別に実行します。

手順

ステップ 1 **show platform**

例：

```
RP/0/RP0/cpu 0: router#show platform
```

XR EXEC モードで **show platform** コマンドを実行すると、さまざまな RP および LC で動作している XR コンソールのステータスが表示されます。

```
RP/0/RP0/CPU0:ios#show platform
Wed Mar 13 22:35:22.679 IST
Node              Type                         State             Config state
--------------------------------------------------------------------------------
0/0/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/1/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/2/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/3/CPU0          A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/4/CPU0          A900-IMA8Z                   OPERATIONAL       NSHUT
0/5/CPU0          A900-IMA8Z                   OPERATIONAL       NSHUT
0/7/CPU0          N560-IMA2C                   OPERATIONAL       NSHUT
0/9/CPU0          N560-IMA2C                   OPERATIONAL       NSHUT
0/10/CPU0         A900-IMA8Z                   OPERATIONAL       NSHUT
0/11/CPU0         A900-IMA8Z                   OPERATIONAL       NSHUT
0/12/CPU0         A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/13/CPU0         A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/14/CPU0         A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/15/CPU0         A900-IMA8CS1Z-M              OPERATIONAL       NSHUT
0/RP0/CPU0        N560-RSP4-E(Active)          IOS XR RUN        NSHUT
0/RP1/CPU0        N560-RSP4-E(Standby)         IOS XR RUN        NSHUT
0/FT0/CPU0        N560-FAN-H                   OPERATIONAL       NSHUT
0/PM0/CPU0        A900-PWR1200-A               OPERATIONAL       NSHUT
0/PM2/CPU0        A900-PWR1200-A               OPERATIONAL       NSHUT
RP/0/RP0/CPU0:ios
```

すべての RP が表示され、それぞれの状態が OPERATIONAL であることを確認します。これは、XR コンソールがカード上で動作していることを示します。

ステップ 2 **admin**

例：

```
RP/0/RP0/cpu 0: router# admin
```

モードを開始します。

ステップ 3 **show platform**

例：

```
sysadmin-vm:0_RP0#show platform
```

システム管理EXECモードで **show platform** コマンドを実行すると、ルータ上のカード（RP、、IM、）、およびハードウェア モジュール（ファン トレイ）などのすべてのハードウェア ユニットのステータスが表示されます。

次に、単一シャーシ システムでの例を示します。

```
sysadmin-vm:0_RP0# show platform
Thu Mar  28 08:19:08.640 UTC+00:00
Location  Card Type                HW State      SW State      Config State
------------------------------------------------------------------------------
0/0       NCS4200-1T16G-PS         OPERATIONAL   N/A           NSHUT
0/1       NCS4200-1T16G-PS         OPERATIONAL   N/A           NSHUT
0/2       NCS4200-1T16G-PS         OPERATIONAL   N/A           NSHUT
0/3       NCS4200-1T16G-PS         OPERATIONAL   N/A           NSHUT
0/4       A900-IMA8Z               OPERATIONAL   N/A           NSHUT
0/5       A900-IMA8Z               OPERATIONAL   N/A           NSHUT
```

```
0/7       N560-IMA2C              OPERATIONAL   N/A           NSHUT
0/9       N560-IMA2C              OPERATIONAL   N/A           NSHUT
0/10      A900-IMA8Z              OPERATIONAL   N/A           NSHUT
0/11      A900-IMA8Z              OPERATIONAL   N/A           NSHUT
0/12      NCS4200-1T16G-PS        OPERATIONAL   N/A           NSHUT
0/13      NCS4200-1T16G-PS        OPERATIONAL   N/A           NSHUT
0/14      NCS4200-1T16G-PS        OPERATIONAL   N/A           NSHUT
0/15      NCS4200-1T16G-PS        OPERATIONAL   N/A           NSHUT
0/RP0     N560-RSP4-E             OPERATIONAL   OPERATIONAL   NSHUT
0/RP1     N560-RSP4-E             OPERATIONAL   OPERATIONAL   NSHUT
0/FT0     N560-FAN-H              OPERATIONAL   N/A           NSHUT
0/PM0     A900-PWR1200-A          OPERATIONAL   N/A           NSHUT
0/PM2     A900-PWR1200-A          OPERATIONAL   N/A           NSHUT

sysadmin-vm:0_RP0#
```

ルータに取り付けられたすべてのカードが結果に表示されていることを確認します。LC/IMおよびRPのソフトウェア ステータス、FTおよび電源モジュールのハードウェア ステータスは、「OPERATIONAL」である必要があります。ハードウェアおよびソフトウェアの各状態を次に示します。

ハードウェアの状態

- OPERATIONAL：カードは正常に動作しており、完全に機能します。
- POWERED_ON：電源がオンで、カードが起動しています。
- FAILED：カードは電源がオンになっていますが、内部障害が発生しています。
- PRESENT：カードはシャットダウン状態です。
- OFFLINE：ユーザによってカードの状態がオフラインに変更されています。診断のためにカードにアクセスできます。

ソフトウェアの状態

- OPERATIONAL：ソフトウェアは正常に動作しており、完全に機能します。
- SW_INACTIVE：ソフトウェアは完全には動作していません。
- FAILED：ソフトウェアは動作していますが、カードに内部障害が発生しています。

ソフトウェア バージョンの確認

ルータには、プリインストールされた Cisco IOS XR ソフトウェアが付属しています。ソフトウェアの最新バージョンがインストールされていることを確認します。新しいバージョンを使用できる場合は、システムアップグレードを実行してください。これにより新しいバージョンのソフトウェアがインストールされ、ルータに最新の機能セットが提供されます。

ルータで実行されている Cisco IOS XR ソフトウェアのバージョンを確認するには、次のタスクを実行します。

手順

show version

例：

```
RP/0/RP0/cpu 0: router# show version
```

ルータにインストールされている各種ソフトウェア コンポーネントのバージョンを表示します。結果には、Cisco IOS XR ソフトウェアとその各種コンポーネントのバージョンが含まれます。

例

次のタスク

結果を確認して、システム アップグレードまたは追加のパッケージ インストールが必要かどうかを特定します。必要な場合は、「システム アップグレードの実行および機能パッケージのインストール （33 ページ）」の章のタスクを参照してください。

ファームウェア バージョンの確認

ルータのさまざまなハードウェア コンポーネントのファームウェアは、インストールされている Cisco IOS XR イメージと互換性がある必要があります。互換性がないと、ルータの誤動作を引き起こす可能性があります。ファームウェア バージョンを確認するには、次のタスクを実行します。

手順

show hw-module fpd

例：

```
RP/0/RP0/CPU0:N560_SYSPSV#show hw-module fpd
Wed Mar 13 22:35:40.387 IST
                                                                FPD Versions
                                                                ================
Location   Card type        HWver FPD device       ATR Status   Running Programd
-------------------------------------------------------------------------------
0/0        NCS4200-1T16G-PS  0.0   IMFPGA               CURRENT    1.76    1.76
0/1        NCS4200-1T16G-PS  0.0   IMFPGA               CURRENT    1.76    1.76
0/2        NCS4200-1T16G-PS  0.0   IMFPGA               CURRENT    1.76    1.76
0/3        NCS4200-1T16G-PS  0.0   IMFPGA               CURRENT    1.76    1.76
0/4        A900-IMA8Z        0.0   IMFPGA               CURRENT   17.02   17.02
0/5        A900-IMA8Z        0.0   IMFPGA               CURRENT   17.02   17.02
0/7        N560-IMA2C        0.0   IMFPGA               CURRENT    3.04    3.04
0/9        N560-IMA2C        0.0   IMFPGA               CURRENT    3.04    3.04
0/10       A900-IMA8Z        0.0   IMFPGA               CURRENT   17.02   17.02
```

```
0/11      A900-IMA8Z         0.0   IMFPGA               CURRENT   17.02   17.02
0/12      NCS4200-1T16G-PS   0.0   IMFPGA               CURRENT    1.76    1.76
0/13      NCS4200-1T16G-PS   0.0   IMFPGA               CURRENT    1.76    1.76
0/14      NCS4200-1T16G-PS   0.0   IMFPGA               CURRENT    1.76    1.76
0/15      NCS4200-1T16G-PS   0.0   IMFPGA               CURRENT    1.76    1.76
0/RP0     N560-RSP4-E        0.0   IOFPGA               CURRENT    0.53    0.53
0/RP0     N560-RSP4-E        0.0   PRIMARY-BIOS         CURRENT    0.14    0.14
0/RP1     N560-RSP4-E        0.0   IOFPGA               CURRENT    0.53    0.53
0/RP1     N560-RSP4-E        0.0   PRIMARY-BIOS         CURRENT    0.14    0.14
0/FT0     N560-FAN-H         0.256 PSOC                 CURRENT    2.01    2.01
0/PM0     A900-PWR1200-A     0.0   PriMCU               NOT READY  0.00    0.00
0/PM0     A900-PWR1200-A     0.0   SecMCU               NOT READY  0.00    0.00
0/PM2     A900-PWR1200-A     0.0   PriMCU               NOT READY  0.00    0.00
0/PM2     A900-PWR1200-A     0.0   SecMCU               NOT READY  0.00    0.00
```

ルータで検出されたハードウェア モジュールのリストを表示します。

（注）　このコマンドは、XR VM とシステム管理 VM の両方のモードで実行できます。

上記の出力で重要なフィールドは次のとおりです。

- FPD Device：IO FPGA、IM FPGA、BIOS などのハードウェア コンポーネントの名前。
- Status：ファームウェアのアップグレード ステータス。それぞれの状態については次のとおりです。
  - CURRENT：ファームウェア バージョンは最新バージョンです。
  - READY：FPD のファームウェアはアップグレード可能な状態です。
  - NOT READY：FPD のファームウェアはアップグレード可能な状態ではありません。
  - NEED UPGD：インストール済みのイメージで新しいファームウェア バージョンを利用できます。アップグレードすることが推奨されます。
  - RLOAD REQ：アップグレードが完了していて、ISOイメージのリロードが必要です。
  - UPGD DONE：ファームウェア アップグレードが正常に行われました。
  - UPGD FAIL：ファームウェア アップグレードが失敗しました。
  - BACK IMG：ファームウェアが破損しています。ファームウェアを再インストールしてください。
  - UPGD SKIP：インストール済みファームウェアのバージョンが、イメージで利用可能なバージョンよりも上位であるため、アップグレードがスキップされました。
- Running：FPD で現在実行中のファームウェアのバージョン。
- Programmd：モジュールにプログラミングされている FPD のバージョン。

次のタスク

- EXEC モードで **upgrade hw-module location all fpd** コマンドを使用して、必要なファームウェアをアップグレードします。個々の FPD を選択して更新することも、すべてをまとめて更新することもできます。FPDアップグレードを有効にするには、ルータの電源を再投入する必要があります。

（注） BIOS と IOFPGA のアップグレードには、新しいバージョンを有効にするためにルータの電源の再投入が必要です。

インターフェイス ステータスの確認

ルータが起動すると、使用可能なすべてのインターフェイスがシステムによって検出されます。インターフェイスが検出されない場合、ユニットの異常を示している可能性があります。検出されたインターフェイスの数を確認するには、次のタスクを実行します。

手順

show ipv4 interface summary

例：

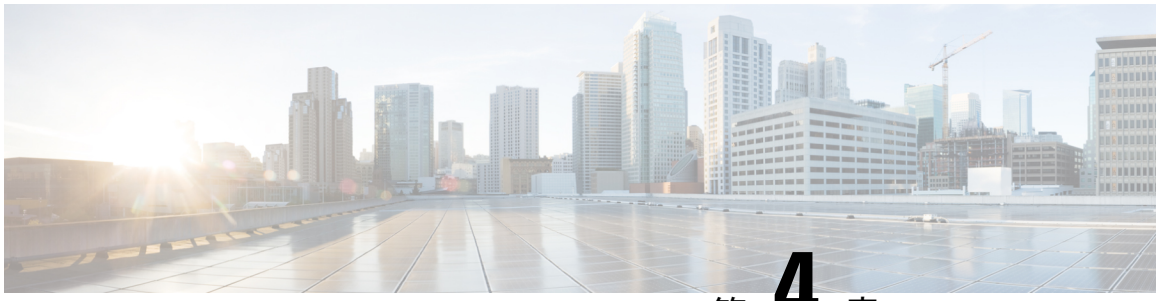
```
RP/0/RP0/cpu 0: router#show ipv4 interface summary
```

ルータの初回起動時には、すべてのインターフェイスが「未割り当て」の状態です。結果に表示されるインターフェイスの総数が、ルータに存在するインターフェイスの実際の数と一致することを確認してください。

上記の結果について説明します。

- Assigned：IP アドレスがインターフェイスに割り当てられています。
- Unnumbered：ルータの他のインターフェイスにすでに設定された IP アドレスを借用しているインターフェイスです。
- Unassigned：IP アドレスはインターフェイスに割り当てられていません。

また、XR EXEC モードで **show interfaces brief** および **show interfaces summary** コマンドを使用し、インターフェイス ステータスを確認することもできます。



第 4 章

ユーザ プロファイルの作成および権限の割り当て

ルータ上のシステム管理設定へのアクセス権を管理するには、権限を割り当てたユーザ プロファイルを作成します。権限はコマンド ルールとデータ ルールを使用して指定します。ユーザ、グループ、コマンド ルール、およびデータ ルールを作成するには、認証、認可、およびアカウンティング（AAA）コマンドをシステム管理コンフィギュレーション モードで使用します。aaa コマンドはディザスタ リカバリ パスワードを変更する際にも使用します。

（注） システム管理 VM から外部 AAA サーバおよびサービスを設定することはできません。その設定は XR VM からのみ実行できます。

ユーザが制御されていないアクセスを行うのを制限するために AAA 認証を設定します。AAA 認証が設定されていない場合、ユーザに割り当てられたグループに関連付けられたコマンドおよびデータ ルールはバイパスされます。IOS-XR ユーザは、ネットワーク設定プロトコル（NETCONF）、Google 定義のリモート プロシージャ コール（gRPC）または任意の YANG ベースのエージェントを介して、IOS-XR 設定への完全な読み取り/書き込みアクセス権を持つことができます。制御されていないアクセスを許可しないようにするには、いずれかの設定を行う前に AAA 認証を有効にします。

（注） XR 上のいずれかのユーザが削除されている場合、ローカル データベースは、システム管理 VM に最初のユーザが存在するかどうかを確認します。

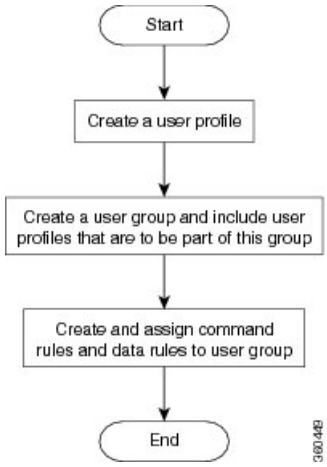
- 最初のユーザが存在する場合、同期は実行されません。
- 最初のユーザが存在しない場合は、XR の最初のユーザ（作成順序に基づく）がシステム管理 VM に同期されます。

ユーザの認証にはユーザ名とパスワードが使用されます。認証されたユーザは、ユーザ グループに対して作成および適用されているコマンド ルールとデータ ルールに基づいて、コマンドを実行しデータ要素にアクセスする権利が与えられます。ユーザ グループに属するすべての

ユーザには、そのユーザ グループのコマンド ルールおよびデータ ルールで定義されているシステムへのアクセス権があります。

ユーザ プロファイルを作成するためのワークフローを次のフローチャートに示します。

図 *1:* ユーザ プロファイル作成のワークフロー



（注） ルータの初回起動時に作成された XR VM の root-lr ユーザは、システム管理 VM の root-system ユーザにマッピングされます。root-system ユーザにはシステム管理 VM のスーパーユーザ権限があるため、アクセスは制限されません。

既存の AAA 設定を表示するには、システム管理コンフィギュレーションconfig モードで **show run aaa** コマンドを使用します。

この章で説明する内容は次のとおりです。

ユーザ プロファイルの作成

システム管理 VM の新しいユーザを作成します。ユーザはユーザ グループに含まれ、特定の権限が割り当てられます。ユーザは割り当てられた権限に基づいて、システム管理 VM コンソールのコマンドと設定への制限付きアクセス権を持ちます。

ルータでは、最大で 1024 個のユーザ プロファイルがサポートされます。

（注） システム管理 VM で作成したユーザは、XR VM で作成したユーザとは異なります。したがって、システム管理 VM ユーザのユーザ名とパスワードを使用して XR VM にアクセスすることはできません。逆も同様です。

XR VM およびシステム管理 VM ユーザ プロファイルの同期

ユーザ プロファイルを XR VM で初めて作成するとき、システム管理 VM にユーザが存在しない場合、ユーザ名とパスワードはシステム管理 VM に同期されます。

ただし、同期されたユーザの XR VM での後続のパスワード変更またはユーザ削除は、システム管理 VM と同期されません。

そのため、XR VM およびシステム管理 VM のパスワードが同じでない可能性があります。また、ユーザが XR VM で削除されても、システム管理 VM と同期されたユーザは削除されません。

XR VM の root-lr ユーザがシステム管理 VM にアクセスするには、XR EXEC モードXR EXEC モードで **Admin** コマンドを入力します。ルータではユーザ名とパスワードの入力を求めるプロンプトは表示されません。XR VM の root-lr ユーザには、システム管理 VM へのフル アクセス権が提供されます。

手順

ステップ 1 **admin**

例：

```
RP/0/RP0/cpu 0: router# admin
```

モードを開始します。

ステップ 2 **config**

例：

```
sysadmin-vm:0_RP0sysadmin-vm:0_RP0#config
```

システム管理コンフィギュレーション System Admin Config モードを開始します。

ステップ 3 **aaa authentication users user** *user_name*

例：

```
sysadmin-vm:0_RP0(config)#aaa authentication users user us1
```

新しいユーザを作成し、ユーザ コンフィギュレーション モードを開始します。例では、ユーザ「us1」が作成されます。

ステップ 4 **password** *password*

例：

```
sysadmin-vm:0_RP0(config-user-us1)#password pwd1
```

システム管理 VM へのログイン時にユーザ認証に使用するパスワードを入力します。

ステップ 5 **uid** *user_id_value*

例：

```
sysadmin-vm:0_RP0(config-user-us1)#uid 100
```

数値を指定します。32 ビットの整数を入力できます。

ステップ 6 **gid** *group_id_value*

例：

```
sysadmin-vm:0_RP0(config-user-us1)#gid 50
```

数値を指定します。32 ビットの整数を入力できます。

ステップ 7 **ssh_keydir** *ssh_keydir*

例：

```
sysadmin-vm:0_RP0(config-user-us1)#ssh_keydir dir1
```

英数字の値を指定します。

ステップ 8 **homedir** *homedir*

例：

```
sysadmin-vm:0_RP0(config-user-us1)#homedir dir2
```

英数字の値を指定します。

ステップ 9 **commit**

次のタスク

- このタスクで作成したユーザを含めるユーザ グループを作成します。ユーザ グループの作成 （24 ページ）を参照してください。
- ユーザ グループに適用するコマンド ルールを作成します。コマンド ルールの作成 （26 ページ）を参照してください。
- ユーザ グループに適用するデータ ルールを作成します。データ ルールの作成 （28 ページ）を参照してください。

ユーザ グループの作成

新しいユーザ グループを作成してコマンド ルールとデータ ルールを関連付けます。コマンド ルールおよびデータ ルールは、ユーザ グループに属するすべてのユーザに適用されます。

ルータでは、最大 32 のユーザ グループがサポートされます。

始める前に

ユーザ プロファイルを作成します。ユーザ プロファイルの作成および権限の割り当て （21 ページ）を参照してください。

手順

ステップ 1 **admin**

例：

```
RP/0/RP0/cpu 0: router# admin
```

モードを開始します。

ステップ 2 **config**

例：

```
sysadmin-vm:0_RP0#config
```

システム管理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 authentication groups group** *group_name*

例：

```
sysadmin-vm:0_RP0(config)#aaa authentication groups group gr1
```

新しいユーザ グループ（まだ存在していない場合）を作成して、グループ コンフィギュレーション モードを開始します。この例では、ユーザ グループ「gr1」が作成されます。

（注） デフォルトで、root ユーザの作成時にユーザ グループ「root-system」がシステムによって作成されます。rootユーザはこのユーザ グループのメンバです。このグループに追加されたユーザは root ユーザ権限を取得します。

ステップ 4 **users** *user_name*

例：

```
sysadmin-vm:0_RP0(config-group-gr1)#users us1
```

ユーザ グループに含めるユーザの名前を指定します。

複数のユーザ名を二重引用符で囲んで指定することができますたとえば、**users** "*user1 user2 ...* "となります。

ステップ 5 **gid** *group_id_value*

例：

```
sysadmin-vm:0_RP0(config-group-gr1)#gid 50
```

数値を指定します。32 ビットの整数を入力できます。

ステップ 6 **commit**

次のタスク

- コマンド ルールを作成します。コマンド ルールの作成 （26 ページ）を参照してください。
- データ ルールを作成します。データ ルールの作成 （28 ページ）を参照してください。

コマンド ルールの作成

コマンド ルールとは、ユーザ グループ内のどのユーザが特定のコマンドの使用を許可または拒否されるかに基づいたルールです。コマンド ルールはユーザ グループに関連付けられ、そのユーザ グループに属するすべてのユーザに適用されます。

コマンドでの動作を許可するか拒否するかを指定することで、コマンドルールを作成します。次の表に、有効な動作と権限の組み合わせを示します。

動作	承認権限	拒否権限
読み取り（R）	「?」を使用した場合にCLIにコマンドが表示されます。	「?」を使用した場合にCLIにコマンドが表示されません。
実行（X）	CLI からコマンドを実行できます。	CLI からコマンドを実行できません。
読み取りおよび実行（RX）	コマンドが CLI に表示され、実行可能です。	コマンドは CLI に表示されず、実行することもできません。

デフォルトでは、すべての権限が **Reject** に設定されています。

各コマンド ルールは、関連付けられている番号によって識別されます。ユーザ グループに複数のコマンド ルールを適用すると、より小さい番号のコマンド ルールが優先されます。たとえば cmdrule 5 は読み取りアクセスを許可しますが、cmdrule 10 は読み取りアクセスを拒否するとします。これら両方のコマンド ルールを同じユーザ グループに適用すると、cmdrule 5 が優先されるため、このグループのユーザは読み取りアクセス権を持ちます。

このタスクの例として、「show platform」コマンドの読み取りおよび実行権限を拒否するルールを作成します。

始める前に

ユーザ グループを作成します。ユーザ グループの作成 （24 ページ）を参照してください。

手順

ステップ 1 **admin**

例：

```
RP/0/RP0/cpu 0: router# admin
```

モードを開始します。

ステップ 2 **config**

例：

```
sysadmin-vm:0_RP0#config
```

システム管理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 authorization cmdrules cmdrule** *command_rule_number*

例：

```
sysadmin-vm:0_RP0(config)#aaa authorization cmdrules cmdrule 1100
```

コマンド ルール番号として数値を指定します。32 ビットの整数を入力できます。

重要　1 ～ 1000 の数字はシスコで予約済みのため使用しないでください。

このコマンドによって、新しいコマンドルール（まだ存在していない場合）が作成され、コマンドルール コンフィギュレーション モードが開始されます。例では、コマンドルール「1100」が作成されます。

（注）　デフォルトでは、root-systemユーザの作成時に「cmdrule 1」がシステムによって作成されます。このコマンドルールは、すべてのコマンドの「読み取り」および「実行」動作に対する「承認」権限を提供します。したがって「cmdrule 1」が変更されない限り、root ユーザに課せられる制限はありません。

ステップ 4 **command** *command_name*

例：

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#command "show platform"
```

権限を制御するコマンドを指定します。

commandにアスタリスク「*****」を入力した場合、そのコマンドルールがすべてのコマンドに適用されることを意味します。

ステップ 5 **ops {r | x | rx}**

例：

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#ops rx
```

権限を指定する必要がある動作を指定します。

- **r** ：読み取り
- **x** ：実行
- **rx** ：読み取りおよび実行

ステップ 6 **action** {**accept** | **accept_log** | **reject**}

例：

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#action reject
```

ユーザがその動作の使用を許可されるか拒否されるかを指定します。

- **accept** ：ユーザはその動作の実行を許可されます。

- **accept_log**：ユーザはその動作の実行を許可され、アクセスの試行がすべて記録されます。
- **reject**：ユーザはその動作の実行を制限されます。

ステップ 7 **group** *user_group_name*

例：

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#group gr1
```

コマンド ルールを適用するユーザ グループを指定します。

ステップ 8 **context** *connection_type*

例：

```
sysadmin-vm:0_RP0(config-cmdrule-1100)#context *
```

このルールを適用する接続タイプを指定します。接続タイプは*netconf*（ネットワーク設定プロトコル）、*cli*（コマンドライン インターフェイス）、または*xml*（Extensible Markup Language）です。アスタリスク「*」の入力が推奨されます。これは、そのコマンド ルールがすべての接続タイプに適用されることを示します。

ステップ 9 **commit**

次のタスク

データ ルールを作成します。データ ルールの作成 （28 ページ）を参照してください。

データ ルールの作成

データ ルールとは、ユーザ グループ内のどのユーザが設定データ要素へのアクセスとその変更を許可または拒否されるかに基づいたルールです。データ ルールはユーザ グループに関連付けられます。データ ルールは、ユーザ グループに属するすべてのユーザに適用されます。

各データ ルールは、関連付けられている番号によって識別されます。ユーザ グループに複数のデータ ルールを適用すると、より小さい番号のデータ ルールが優先されます。

始める前に

ユーザ グループを作成します。ユーザ グループの作成 （24 ページ）を参照してください。

手順

ステップ 1 **admin**

例：

```
RP/0/RP0/cpu 0: router# admin
```

モードを開始します。

ステップ 2 **config**

例：

```
sysadmin-vm:0_RP0#config
```

システム管理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 authorization datarules datarule** *data_rule_number*

例：

```
sysadmin-vm:0_RP0(config)#aaa authorization datarules datarule 1100
```

データ ルール番号として数値を指定します。32 ビットの整数を入力できます。

重要 1 ～ 1000 の数字はシスコで予約済みのため使用しないでください。

このコマンドによって、新しいデータ ルール（まだ存在していない場合）が作成され、データ ルール コンフィギュレーション モードが開始されます。例では、データ ルール「1100」が作成されます。

（注） デフォルトで、root-system ユーザの作成時に「datarule 1」がシステムによって作成されます。このデータ ルールは、すべての設定データの「読み取り」、「書き込み」、および「実行」動作に対する「承認」権限を提供します。したがって「datarule 1」が変更されない限り、root ユーザに課せられる制限はありません。

ステップ 4 **keypath** *keypath*

例：

```
sysadmin-vm:0_RP0(config-datarule-1100)#keypath  /aaa/disaster-recovery
```

データ要素のキーパスを指定します。キーパスはデータ要素の場所を定義する式です。**keypath** にアスタリスク「*」を入力した場合、そのコマンド ルールがすべての設定データに適用されることを意味します。

ステップ 5 **ops** *operation*

例：

```
sysadmin-vm:0_RP0(config-datarule-1100)#ops rw
```

権限を指定する必要がある動作を指定します。各動作は次の文字で識別されます。

- c：作成
- d：削除
- u：更新
- w：書き込み（作成、更新、および削除の組み合わせ）
- r：読み込み
- x：実行

ステップ 6 **action** {**accept** | **accept_log** | **reject**}

例：

```
sysadmin-vm:0_RP0(config-datarule-1100)#action reject
```

ユーザがその動作を許可されるか拒否されるかを指定します。

- **accept** ：ユーザはその動作の実行を許可されます。
- **accept_log**：ユーザはその動作の実行を許可され、アクセスの試行がすべて記録されます。
- **reject**：ユーザはその動作の実行を制限されます。

ステップ 7 **group** *user_group_name*

例：

```
sysadmin-vm:0_RP0(config-datarule-1100)#group gr1
```

データ ルールを適用するユーザ グループを指定します。複数のグループ名を指定することもできます。

ステップ 8 **context** *connection type*

例：

```
sysadmin-vm:0_RP0(config-datarule-1100)#context *
```

このルールを適用する接続タイプを指定します。接続タイプは*netconf*（ネットワーク設定プロトコル）、*cli*（コマンドライン インターフェイス）、または*xml*（Extensible Markup Language）です。アスタリスク「*」の入力が推奨されます。これは、そのコマンドがすべての接続タイプに適用されることを示します。

ステップ 9 **namespace** *namespace*

例：

```
sysadmin-vm:0_RP0(config-datarule-1100)#namespace *
```

アスタリスク「*」を入力して、データ ルールが名前空間の値すべてに適用されることを示します。

ステップ 10 **commit**

ディザスタ リカバリのユーザ名とパスワードの変更

ルータの起動後、最初にroot-systemユーザ名とパスワードを定義すると、同じユーザ名とパスワードがシステム管理コンソールのディザスタ リカバリ ユーザ名およびパスワードとしてマッピングされます。ただし、これらは変更可能です。

ディザスタ リカバリ ユーザ名およびパスワードは、次の状況で役立ちます。

- システム管理コンソールでの認証のデフォルト ソースである AAA データベースが破損した場合にシステムへアクセスする。

- 何らかの理由でシステム管理コンソールが機能しない場合に、管理ポートを通じてシステムにアクセスする。
- 通常のユーザ名およびパスワードを忘れた場合に、ディザスタ リカバリ ユーザ名とパスワードを使用してシステム管理コンソールにアクセスし、新しいユーザを作成する。

（注） ルータでは、ディザスタ リカバリ ユーザ名およびパスワードを一度に1つのみ設定できます。

手順

ステップ 1 **admin**

例：

```
RP/0/RP0/cpu 0: router# admin
```

モードを開始します。

ステップ 2 **config**

例：

```
sysadmin-vm:0_RP0#config
```

システム管理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 disaster-recovery username** *username* **password** *password*

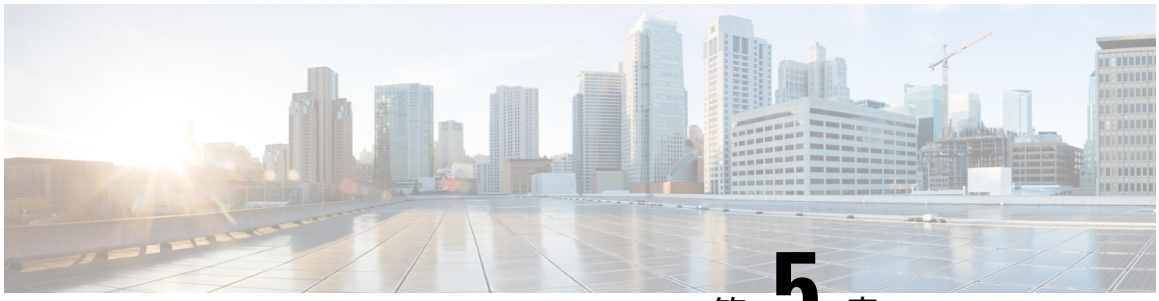
例：

```
sysadmin-vm:0_RP0(config)#aaa disaster-recovery username us1 password pwd1
```

ディザスタ リカバリ ユーザ名とパスワードを指定します。既存のユーザをディザスタ リカバリ ユーザとして選択する必要があります。この例では、ディザスタ リカバリ ユーザとして「us1」が選択され、パスワード「pwd1」が割り当てられます。パスワードは、プレーンテキストまたは MD5 ダイジェスト文字列として入力することができます。

ディザスタ リカバリ ユーザ名を使用する場合は、*username***@localhost** の形式で入力してください。

ステップ 4 **commit**



第 5 章

システム アップグレードの実行および機能パッケージのインストール

システム アップグレードおよびパッケージ インストール プロセスを実行するには、ルータで **install** コマンドを使用します。これらのプロセスでは、iso イメージ（*.iso*）、機能パッケージ（*.rpm*）、およびソフトウェア メンテナンス アップグレード ファイル（*.smu*）をルータ上で追加およびアクティブ化します。ネットワーク サーバからこれらのファイルにアクセスし、ルータ上でアクティブ化します。インストールしたパッケージまたは SMU が原因でルータに問題が発生した場合は、アンインストールすることができます。

この章で説明する内容は次のとおりです。

システムのアップグレード

システムのアップグレードとは、ルータに新しいバージョンの Cisco IOS XR オペレーティング システムをインストールするプロセスです。ルータには Cisco IOS XR イメージがプリインストールされています。ただし、ルータ機能を最新の状態に保つために新しいバージョンをインストールすることができます。システム アップグレードの操作は XR VM から実行しますが、システム アップグレード時に、XR VM とシステム管理 VM の両方で動作しているオペレーティング システムがアップグレードされます。

NCS-55A2-MOD-SL、NCS-55A2-MOD-HD-S、NCS-55A2-MOD-HX-S、NCS-55A2-MOD-SE-S のいずれかの NCS 55A2 固定シャーシをリロードすると、1G インターフェイスはモジュラ ポート アダプタ（MPA）NC55-MPA-12T-S 内で 1 回ではなく 2 回フラッピングします。

（注） ルータ上のインターフェイスに設定が行われておらず、no-shut操作を実行して起動した場合、ルータのリロード時にインターフェイスの状態が自動的に **admin-shutdown** に変更されます。

システム アップグレードは、基本パッケージ（Cisco IOS XR ユニキャスト ルーティング コア バンドル）のインストールによって行います。このバンドルのファイル名は *ncs560-mini-x.iso* です。**install** コマンドを使用して、このISOイメージをインストールします。インストールプロセスの詳細については、インストール プロセスのワークフロー （35 ページ）を参照してください。

注意 ルータのリロード時はインストール操作を実行しないでください。

アップグレード操作中はルータをリロードしないでください。

Cisco IOS XR は、ISO およびアップグレードイメージで Cisco IOS XR RPM パッケージの RPM 署名と署名検証をサポートしています。Cisco IOS XR ISO およびアップグレードイメージのすべての RPM パッケージは、暗号化の完全性と信頼性を保証するために署名されます。これにより、RPM パッケージが改ざんされておらず、RPM パッケージが Cisco IOS XR からのものであることが保証されます。RPM パッケージの署名に使用される秘密キーは、シスコによって作成され、安全に維持されます。

システムと RPM のアップグレードの詳細については、「自動依存関係管理」の章を参照してください。

機能のアップグレード

機能のアップグレードとは、ルータに新機能とソフトウェアパッチを導入するプロセスです。機能アップグレードは、パッケージファイル（単にパッケージと呼ばれます）のインストールによって行います。ソフトウェア パッチのインストールはソフトウェア メンテナンス アップグレード（SMU）ファイルのインストールによって行います。

ルータにパッケージをインストールすると、そのパッケージに含まれる特定の機能がインストールされます。Cisco IOS XR ソフトウェアはさまざまなソフトウェア パッケージに分割されているため、ルータで実行する機能を選択することができます。各パッケージには、ルーティングやセキュリティなど、特定のルータ機能のセットを実行するコンポーネントが含まれています。

たとえばルーティング パッケージのコンポーネントは、BGP や OSPF など、個別の RPM に分かれています。BGP は必須 RPM であり、基本ソフトウェア バージョンに含まれているので削除できません。OSPF などの任意の RPM は、必要に応じて追加および削除できます。

パッケージの命名規則は `<platform>-<pkg>-<pkg version>-<release version>.<architecture>.rpm` です。標準パッケージは次のとおりです。

- ncs560-mpls-1.0.0.0-<リリース番号>.x86_64.rpm

- ncs560-isis-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-mcast-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-mgbl-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-bgp-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-ospf-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-mpls-te-rsvp-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-li-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-eigrp-1.0.0.0-<リリース番号>.x86_64.rpm
- ncs560-k9sec-1.0.0.0-<リリース番号>.x86_64.rpm

パッケージおよびSMUのインストールは、**install**コマンドを使用して実行します。インストール プロセスの詳細については、パッケージのインストール （35 ページ）を参照してください。

XR VM とシステム管理 VM 用の個別のパッケージおよび SMU があります。それぞれをそのファイル名で識別できます。

システムのアップグレードおよび RPM の詳細については、『*Cisco IOS XR Flexible Packaging Configuration Guide*』を参照してください。

インストール プロセスのワークフロー

インストールおよびアンインストールプロセスのワークフローについては、次のフローチャートを参照してください。

パッケージのインストールについては、パッケージのインストール （35 ページ）を参照してください。パッケージのアンインストールについては、パッケージのアンインストール （43 ページ）を参照してください。

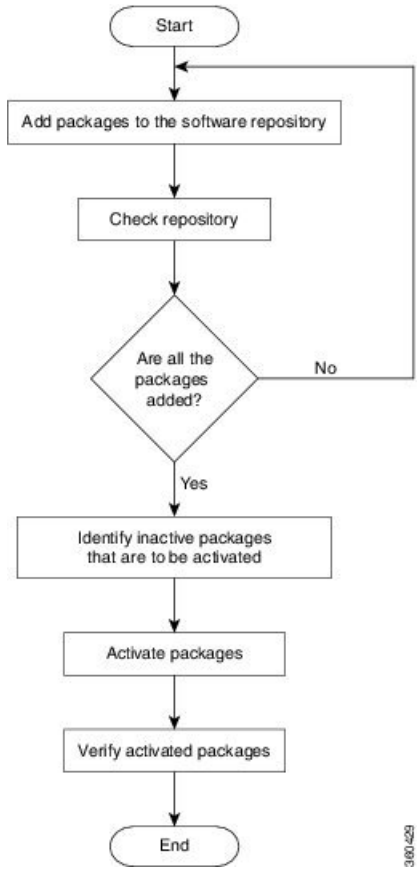
パッケージのインストール

システムをアップグレードするか、パッチをインストールするには、このタスクを完了します。システム アップグレードは ISO イメージ ファイルを使用して行いますが、パッチ インストールの場合はパッケージおよび SMU を使用します。*.rpm* ファイルをインストールする際もこのタスクを使用します。*.rpm* ファイルには、1 つのファイルに統合された複数のパッケージと SMU が含まれています。カード タイプにかかわらず、パッケージ形式によってコンポーネントごとに 1 つの RPM が定義されます。

（注） システム管理パッケージおよびXRパッケージは、システム管理EXECモードおよびXR EXECモードで**install**コマンドを使用して実行できます。すべての**install**コマンドは両方のモードで使用できます。

パッケージをインストールするためのワークフローを次のフローチャートに示します。

図 *2:* パッケージ インストールのワークフロー



始める前に

- 管理ポートを設定して接続します。インストール可能なファイルには管理ポートからアクセスできます。管理ポートの設定の詳細については、管理ポートの設定 （11 ページ） を参照してください。
- インストールするパッケージを、ルータのハードディスク、またはルータがアクセスできるネットワーク サーバにコピーします。

手順

ステップ 1 次のいずれかを実行します。

- **install add source** *<tftp transfer protocol>/package_path/ filename1 filename2 ...*
- **install add source** *<ftp or sftp transfer protocol>//user@server:/package_path/ filename1 filename2 ...*

例：

```
RP/0/RP0/CPU0:router#install add source
/harddisk:/ncs560-mpls-te-rsvp-1.0.0.0-r662.x86_64.rpm
ncs560-mgbl-1.0.0.0-r662.x86_64.rpm
```

または

```
RP/0/RP0/CPU0:router#install add source tftp://root@8.33.5.15:/auto/ncs/package
```

または

```
RP/0/RP0/CPU0:router#install add source
tftp://root@8.33.5.15:/auto/ncs/package/ncs560-mcast-1.0.0.0-r662.x86_64.rpm
 ncs560-mpls-1.0.0.0-r662.x86_64.rpm
```

（注） *package_path* と *filename* の間にはスペースが必要です。

パッケージからソフトウェアファイルが展開され、ソフトウェアリポジトリに追加されます。追加するファイルのサイズによっては、この処理に時間がかかる場合があります。動作は非同期モードで実行されます。**install add** コマンドはバックグラウンドで実行され、EXEC プロンプトは最短で返されます。

（注） XR VM とシステム管理 VM のリポジトリは異なります。ルーティング パッケージは XR VM リポジトリに、システム管理パッケージはシステム管理 VM リポジトリに自動的に追加されます。

ステップ 2 **show install request**

例：

```
RP/0/RP0/CP0:ios# show install request

Thu Mar 28 13:29:03.219 IST

The install add operation 36 is 30% complete
RP/0/RP0/CPU0:ios#
```

（任意）追加動作の動作 ID とステータスを表示します。動作 ID は、後で **activate** コマンドを実行する際に使用できます。

```
Install operation 8 is still in progress
```

システム管理パッケージの場合は、残りの手順をシステム管理 EXEC モードで実行する必要があります。システム管理 EXEC モードを開始するには、**admin** コマンドを使用します。

ステップ 3 **show install repository**

例：

```
RP/0/RP0/CPU0:ios# show install repository all
Thu Mar 28 13:58:40.796 IST
1 package(s) in Host repository:
    host-6.6.2
2 package(s) in Admin repository:
    ncs560-mini-x-6.6.2
    ncs560-sysadmin-7.0.1.128I
10 package(s) in XR repository:
    ncs560-xr-6.6.2
    ncs560-mgbl-2.0.0.0-r662.x86_64
    ncs560-mini-x-6.6.2
    ncs560-isis-2.0.0.0-r662.x86_64
    ncs560-k9sec-2.0.0.0-r662.x86_64
    ncs560-mpls-te-rsvp-2.0.0.0-r662.x86_64
    ncs560-mpls-1.0.0.0-r662.x86_64
    ncs560-mcast-2.0.0.0-r662.x86_64
    ncs560-li-1.0.0.0-r662.x86_64
    ncs560-ospf-2.0.0.0-r662.x86_64
RP/0/RP0/CPU0:ios#
```

リポジトリに追加されるパッケージを表示します。パッケージは `install add` 動作の完了後にのみ表示されます。

ステップ 4 **show install inactive**

例：

```
RP/0/RP0/CPU0:ios# show install inactive
```

リポジトリ内に存在する非アクティブなパッケージを表示します。アクティブ化できるのは非アクティブなパッケージだけです。

ステップ 5 次のいずれかを実行します。

- **install activate** *package_name*
- **install activate id** *operation_id*

例：

```
RP/0/RP0/CPU0:router#install activate ncs560-mcast-1.0.0.0-r662.x86_64.rpm
ncs560-mpls-1.0.0.0-r662.x86_64.rpm
```

operation_id は **install add** 動作の ID です。このコマンドは、システム管理モードでも実行できます。パッケージ設定がルータでアクティブになります。その結果、新機能とソフトウェア修正が有効になります。この動作は非同期モードで実行されます。**install activate** コマンドはバックグラウンドで実行され、EXEC プロンプトが返されます。

動作 ID を使用すると、指定した動作に追加されたすべてのパッケージがまとめてアクティブ化されます。たとえば 5 つのパッケージが動作 8 に追加されている場合、**install activate id 8** を実行すると、5 つのパッケージがすべてまとめてアクティブ化されます。パッケージを個別にアクティブ化する必要はありません。

アクティベーションは瞬時には完了せず、ある程度の時間がかかります。SMU によっては、アクティベーション時にルータの手動リロードが必要な場合があります。このような SMU をアクティブ化すると、リロードを実行するための警告メッセージが表示されます。SMU のコンポーネントは、リロードの完了後にのみアクティブ化されます。**install activate** コマンドの実行後すぐにルータをリロードします。SMU が XR VM とシステム管理 VM の両方と依存関係がある場合は、両方の VM で SMU をアクティブ化した後でリロードを実行すると、両方で

同時に反映されます。ルータをリロードするには、システム管理EXECモードから**hw-module location all reload** コマンドを使用します。

ステップ 6 **show install active**

例：

```
RP/0/RP0/CPU0:ios# show install active
```

アクティブなパッケージを表示します。

ステップ 7 **install commit**

例：

```
RP/0/RP0/CPU0:ios# install commit
```

XR の新たにアクティブ化されたソフトウェアをコミットします。XR とシステム管理ソフトウェアの両方をコミットするには、**install commit system** を使用します。

パッケージのインストール：関連コマンド

関連コマンド	目的
show install log	インストールプロセスのログ情報を表示します。これはインストールが失敗した場合のトラブルシューティングに使用できます。
show install package	リポジトリに追加されたパッケージの詳細を表示します。このコマンドは、パッケージの個々のコンポーネントを識別する際に使用します。
install prepare	アクティベーションの準備として、非アクティブなパッケージに対してアクティベーション前のチェックを実行します。
show install prepare	準備が完了してアクティベーション可能になったパッケージのリストを表示します。

次のタスク

- システムのアップグレードの実行後、システム管理 EXEC モードから **upgrade hw-module location all fpd all** コマンドを使用して FPD をアップグレードします。FPD アップグレード プロセスの進捗状況は、システム管理 EXEC モードで **show hw-module fpd** コマンドを使用してモニタできます。FPDアップグレードが完了したら、ルータをリロードします。
- **install verify packages** コマンドを使用してインストールを確認します。
- インストールによってルータに問題が発生した場合は、該当するパッケージまたは SMU をアンインストールしてください。パッケージのアンインストール （43 ページ）を参照してください。

（注） ISOイメージはアンインストールできません。ただし、旧バージョンのISOをインストールすることでシステムダウングレードを実行することができます。

（注） 電源モジュールをアップグレードする場合は、最初にSysadminプロンプトから **upgrade hw-module location <SC0/SC1> fpd all** コマンドを使用して SC IO FPGA をアップグレードし、次に **upgrade hw-module location pm-all fpd** コマンドを使用して FPD をアップグレードしてください。

最後に、Sysadmin プロンプトから **hw-module location <SC0/SC1> reload** コマンドを使用してシェルフコントローラをリロードします。

準備済みパッケージのインストール

システムアップグレードまたは機能アップグレードは、ISOイメージファイル、パッケージ、および SMU をアクティブ化することで実行します。アクティベーション前にこれらのインストール可能なファイルを準備することができます。準備フェーズでは、アクティベーション前のチェックが行われ、インストール可能なファイルのコンポーネントがルータ設定にロードされます。準備プロセスはバックグラウンドで実行されるため、その間もルータをフルに利用できます。準備フェーズが完了したら、すべての準備済みファイルを即座にアクティブ化できます。アクティベーション前の準備には、次の利点があります。

- インストール可能なファイルが破損していると、準備プロセスは失敗します。これによって問題が早期に警告されます。破損したファイルが直接アクティブ化されると、ルータの誤動作を招く可能性があります。
- システム アップグレード用の ISO イメージを直接アクティブ化するには時間がかかり、その間にルータを使用できなくなります。ただし、アクティベーション前にイメージを準備すると、準備プロセスが非同期で実行されるだけでなく、準備済みのイメージを後でアクティブ化するときに、アクティベーション プロセスにかかる時間も著しく短縮されます。その結果、ルータのダウンタイムが大幅に削減されます。

システムのアップグレードおよびパッケージのインストールに準備動作を利用するには、次のタスクを実行します。

（注） システム管理パッケージまたはXRパッケージのどちらをインストールするかによって、それぞれシステム管理EXECモードまたはXR EXECモードで**install**コマンドを実行します。すべての**install**コマンドは両方のモードで使用できます。システム管理のインストール動作はXRモードで実行できます。

始める前に

- インストール可能なファイルが破損していると、準備プロセスは失敗します。これによって問題が早期に警告されます。破損したファイルが直接アクティブ化されると、ルータの誤動作を招く可能性があります。
- システム アップグレード用のISOイメージを直接アクティブ化するには時間がかかり、その間にルータを使用できなくなります。ただし、アクティベーション前にイメージを準備すると、準備プロセスが非同期で実行されるだけでなく、準備済みのイメージを後でアクティブ化するときに、アクティベーション プロセスにかかる時間も著しく短縮されます。その結果、ルータのダウンタイムが大幅に削減されます。

手順

ステップ1 必要なISOイメージおよびパッケージをリポジトリに追加します。

詳細については、パッケージのインストール （35ページ）を参照してください。

ステップ2 **show install repository**

例：

```
RP/0/RP0/cpu 0: router#show install repository
```

必要なインストール可能ファイルがリポジトリ内にあることを確認するには、この手順を実行します。パッケージは「install add」動作の完了後にのみ表示されます。

ステップ3 次のいずれかを実行します。

- **install prepare** *package_name*
- **install prepare id** *operation_id*

例：

準備プロセスが開始されます。この動作は非同期モードで実行されます。**install prepare**コマンドはバックグラウンドで実行され、EXECプロンプトは最短で返されます。

動作IDを使用すると、指定した動作に追加されたすべてのパッケージの準備がまとめて行われます。たとえば5つのパッケージが動作8に追加されている場合、**install prepare id 8**を実行すると、5つのパッケージがすべてまとめて準備されます。パッケージを個別に準備する必要はありません。

ステップ4 **show install prepare**

例：

```
RP/0/RP0/cpu 0: router#show install prepare
```

準備済みのパッケージを表示します。この結果で、必要なすべてのパッケージが準備されていることを確認します。

ステップ 5 **install activate**

例：

```
RP/0/RP0/cpu 0: router#install activate
```

準備の完了したすべてのパッケージをまとめてアクティブ化し、ルータでパッケージ設定をアクティブにします。

（注） CLI でパッケージ名または動作 ID を指定しないでください。

SMU によっては、アクティベーション時にルータの手動リロードが必要な場合があります。このような SMU をアクティブ化すると、リロードを実行するための警告メッセージが表示されます。SMU のコンポーネントは、リロードの完了後にのみアクティブ化されます。**install activate** コマンドの完了後すぐにルータのリロードを実行します。

ステップ 6 **show install active**

例：

```
RP/0/RP0/cpu 0: router#show install active
```

アクティブなパッケージを表示します。

この結果で、すべての RP と LC でイメージおよびパッケージの同じバージョンがアクティブになっていることを確認します。

ステップ 7 **install commit**

例：

```
RP/0/RP0/cpu 0: router#install commit
```

パッケージのインストール：関連コマンド

関連コマンド	目的
show install log	インストールプロセスのログ情報を表示します。これはインストールが失敗した場合のトラブルシューティングに使用できます。
show install package	リポジトリに追加されたパッケージの詳細を表示します。このコマンドは、パッケージの個々のコンポーネントを識別する際に使用します。
install prepare clean	準備動作をクリアし、すべてのパッケージを準備済み状態から削除します。

次のタスク

- システムのアップグレードの実行後、システム管理システム管理EXECモードから**upgrade hw-module location all fpd all**コマンドを使用してFPDをアップグレードします。FPDアップグレードプロセスの進捗状況は、システム管理 システム管理 EXEC モードで**show hw-module fpd**コマンドを使用してモニタできます。FPDアップグレードが完了したら、ルータをリロードします。
- **install verify packages**コマンドを使用してインストールを確認します。
- インストールによってルータに問題が発生した場合は、該当するパッケージまたはSMUをアンインストールしてください。パッケージのアンインストールを参照してください。

（注） ISOイメージはアンインストールできません。ただし、旧バージョンのISOをインストールすることでシステムダウングレードを実行することができます。

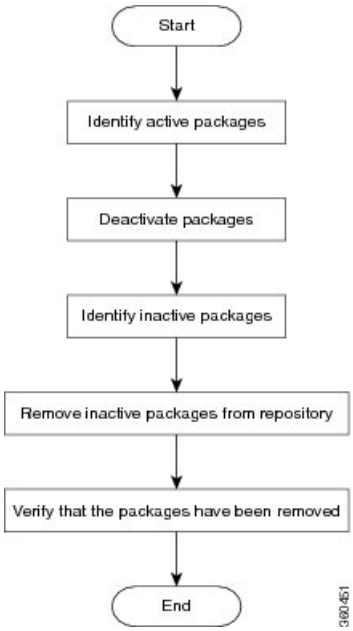
パッケージのアンインストール

パッケージをアンインストールするには、次のタスクを実行します。アンインストールしたパッケージに含まれるすべてのルータ機能は非アクティブ化されます。XR VMで追加したパッケージをシステム管理 VM からアンインストールすることはできません。逆も同様です。

（注） インストール済みの ISO イメージはアンインストールできません。また、ホスト、XR VM、およびシステム管理 VM でサードパーティ製 SMU をインストールするカーネル SMU もアンインストールできません。ただし、ISO イメージまたはカーネル SMU を新たにインストールすると既存のインストールが上書きされます。

パッケージをアンインストールするためのワークフローを次のフローチャートに示します。

図 *3:* パッケージ アンインストールのワークフロー



このタスクでは、XR VM パッケージをアンインストールします。システム管理パッケージをアンインストールする場合は、同じコマンドを システム管理 EXEC モードで実行します。

手順

ステップ 1 **show install active**

例：

```
RP/0/RP0/cpu 0: router#show install active
```

アクティブなパッケージを表示します。非アクティブ化できるのはアクティブなパッケージだけです。

ステップ 2 次のいずれかを実行します。

- **install deactivate** *package_name*
- **install deactivate id** *operation_id*

例：

operation_id は **install add** 動作の ID です。パッケージに関連するすべての機能およびソフトウェア パッチが非アクティブ化されます。複数のパッケージ名を指定して同時に非アクティブ化できます。

動作 ID を使用すると、指定した動作に追加されたすべてのパッケージがまとめて非アクティブ化されます。パッケージを個別に非アクティブ化する必要はありません。**install add** 動作（非アクティブ化で使用した ID の動作）の一部として追加されたシステム管理パッケージがある場合、これらも非アクティブ化されます。

ステップ 3 **show install inactive**

例：

```
RP/0/RP0/cpu 0: router#show install inactive
```

非アクティブ化済みのパッケージは、非アクティブなパッケージとして表示されるようになります。非アクティブなパッケージのみリポジトリから削除できます。

ステップ 4 **install remove** *package_name*

例：

非アクティブなパッケージがリポジトリから削除されます。

指定した動作 ID に追加されているすべてのパッケージを削除するには、**id** *operation-id* キーワードおよび引数を指定して **install remove** コマンドを使用します。

ステップ 5 **show install repository**

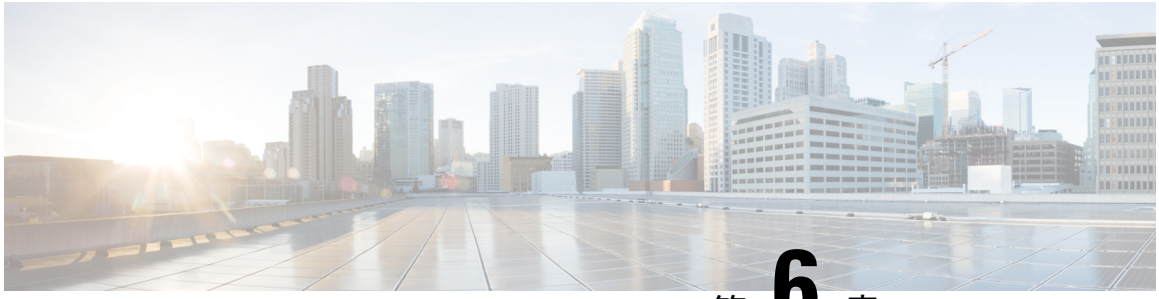
例：

```
RP/0/RP0/cpu 0: router#show install repository
```

リポジトリ内の使用可能なパッケージを表示します。削除されたパッケージは結果に表示されなくなります。

次のタスク

必要なパッケージをインストールします。パッケージのインストール （35 ページ） を参照してください。

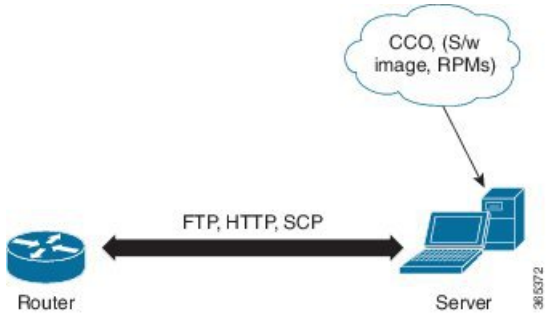


第 6 章

自動依存関係管理

フレキシブル パッケージでは、自動依存関係管理がサポートされます。RPM の更新中に、関連するすべての依存パッケージがシステムによって自動的に特定され、更新されます。

図 *4:* インストール フロー（基本ソフトウェア、*RPM*、および *SMU*）



このリリースまでは、ユーザはネットワーク サーバ（リポジトリ）上の CCO からソフトウェア イメージおよび必要な RPM をダウンロードしていました。また、**install add** コマンドおよび **install activate** コマンドを実行して、ダウンロードしたファイルを に追加し、それらでアクティブ化していました。さらに、関連する依存 RPM を手動で特定し、追加およびアクティブ化する必要がありました。

自動依存関係管理を使用すると、ユーザ側で依存 RPM を特定し、個別に追加およびアクティブ化する必要がありません。また、依存 RPM を自動的に特定し、インストールする新しいインストール コマンドを実行できます。

新しいコマンドは **install update install source** および **install upgrade** です。**install update install source** コマンドは、依存パッケージを特定し、更新します。このコマンドは、基本パッケージを更新しません。 **install upgrade** コマンドは、基本パッケージをアップグレードします。

これ以降で説明する内容は、次のとおりです。

- RPM と SMU の更新 （48 ページ）
- 基本ソフトウェア バージョンのアップグレード （49 ページ）

RPM と SMU の更新

RPM には特定の不具合に対する修正が含まれており、その修正でシステムを更新する必要があります。RPM および SMU を新しいバージョンに更新するには、**install update install source** コマンドを使用します。特定の RPM に対して **install update install source** コマンドが発行されると、 ルータによりリポジトリとの間で通信が行われ、RPM がダウンロードされてアクティブ化されます。依存関係にある RPM がリポジトリにある場合、 ルータによってその RPM が特定され、インストールされます。

install update install source コマンドの構文は次のとおりです。

install update source install source　リポジトリ　**[rpm]**

install update install source コマンドは、次の 4 つの方法で実行できます。

- **パッケージ名を指定しない。**

  パッケージ名を指定しないと、すべてのインストール済みパッケージがコマンドによって最新の SMU で更新されます。

  ```
  install update source install source [repository]
  ```

- **パッケージ名を指定する。**

  パッケージ名を指定すると、そのパッケージがコマンドによってインストールされ、依存関係とともにそのパッケージの最新の SMU で更新されます。パッケージがすでにインストールされている場合、そのパッケージの SMU だけがインストールされます（すでにインストールされている SMU は、スキップされます）。

  ```
  install update source install source [repository] ncs560-mpls.rpm
  ```

- **パッケージ名とバージョン番号を指定する。**

  パッケージの特定のバージョンをインストールする必要がある場合、完全なパッケージ名を指定します。このパッケージは、リポジトリにあるパッケージの最新の SMU とともにインストールされます。

  ```
  install update source install source [repository]
  ncs560-mpls-1.0.2.0-r662.x86_64.rpm
  ```

- **SMU を指定する。**

  SMU を指定すると、その SMU は依存関係にある SMU とともにダウンロードおよびインストールされます。

  ```
  install update source install source [repository]
  ncs560-mpls-1.0.2.1-r662.CSCub12345.x86_64.rpm
  ```

基本ソフトウェア バージョンのアップグレード

基本ソフトウェアは、新しいバージョンが利用可能になった場合に、そのバージョンにアップグレードできます。基本ソフトウェアを最新バージョンにアップグレードするには、**install upgrade** コマンドを使用します。ベース バージョンをアップグレードすると、ルータで現在利用可能な RPM もアップグレードされます。

（注） SMU は、このプロセスの一部としてアップグレードされません。

install upgrade コマンドの構文は次のとおりです。

install upgrade source *repository* **version** *version***[rpm]**

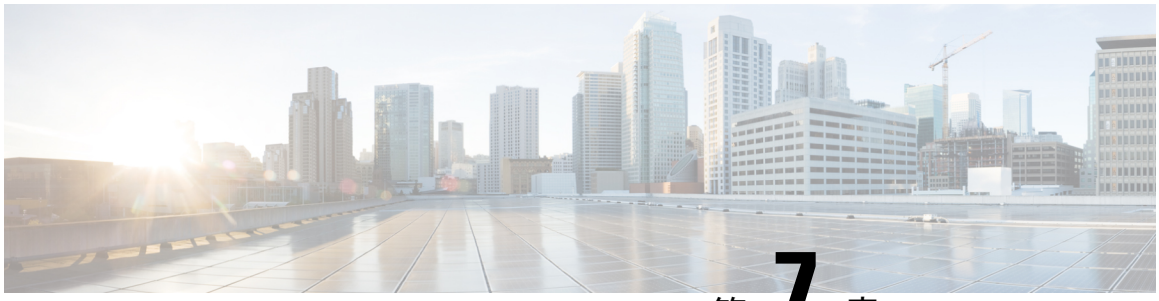
（注） データポート上の VRF および TPA はサポートされません。デフォルト以外の VRF インターフェイスを介してしかサーバにアクセスできない場合、ファイルは、ftp、sfp、scp、http、または https プロトコルを使用してすでに取得しておく必要があります。

install upgrade コマンドを使用するのは次の場合です。

- **バージョン番号を指定する**

  基本ソフトウェア（.mini）は、特定のバージョンにアップグレードされます。すべてのインストール済み RPM も、同じリリース バージョンにアップグレードされます。

  install upgrade source[repository] version *<release-number>*



第 7 章

ゴールデン ISO ワークフロー

次の図は、ゴールデン ISO を構築してインストールするためのワークフローを示しています。

ゴールデン ISO の構築

カスタマイズした ISO は、github の場所（https://github.com/ios-xr/gisobuild）で利用可能なシスコ ゴールデン ISO（GISO）作成スクリプト `gisobuild.py` を使用して構築します。

GISO 作成スクリプトは、自動依存関係管理をサポートし、次の機能を提供します。

- パッケージ リポジトリ内に存在するすべてのパッケージの RPM データベースを構築します。
- mini-x.iso バージョンと一致しない Cisco RPM をスキップおよび削除します。
- mini-x.iso 内にすでに存在するサードパーティ製の基本パッケージの SMU ではないサードパーティの RPM をスキップおよび削除します。
- 同じリリースで異なるバージョンの基本 RPM が複数ある場合、エラーを表示し、作成プロセスを終了します。
- すべての RPM の互換性チェックと依存関係チェックを実行します。たとえば、子 RPM は親 RPM に依存します。子 RPM のみが含まれる場合、ゴールデン ISO の作成は失敗します。

GISO を作成するには、スクリプトに次の入力パラメータを指定します。

- 基本 mini-x.iso（必須）
- XR コンフィギュレーション ファイル（任意）
- ホスト、XR、およびシステム管理用の 1 つまたは複数のシスコ固有の SMU（必須）
- ホスト、XR、およびシステム管理用の 1 つまたは複数のサードパーティ SMU（必須）

- ゴールデン ISO のラベル（任意）

（注） ゴールデン ISO はミニ ISO からのみ作成できます。full または fullk9 バンドル ISO はサポートされていません。

GISO を作成する場合は、次の命名規則を使用します。

GISO ビルド	書式	例
k9sec RPM を使用しない GISO	`<platform-name>-golden-x.iso-<version>.<label>` `<platform-name>-golden-x-<version>.iso.<label>`	<プラットフォーム名>-golden-x64.iso-<バージョン>.v1 <プラットフォーム名>-golden-x64-<バージョン>.iso.v1
k9sec RPM を使用した GISO	`<platform-name>-goldenk9-x.iso-<version>.<label>` `<platform-name>-goldenk9-x-<version>.iso.<label>`	<プラットフォーム名>-goldenk9-x64.iso-<バージョン>.v1 <プラットフォーム名>-goldenk9-x64-<バージョン>.iso.v1

（注） k9sec RPM を GISO に適切に追加するには、**chmod** コマンドを使用してファイルの権限を `644` に変更します。

```
chmod 644 [k9 sec rpm]
```

GISO を作成するには、次の手順を実行します。

始める前に

- 非 GISO から GISO バージョンにアップグレードするには、最初に GISO サポートを使用して ミニ ISO にアップグレードする必要があります。
- GISO が構築されているシステムは、次の要件を満たしている必要があります。
  - システムには Python バージョン 2.7 以降が必要です。
  - システムには、最低 3 ～ 4 GB の空きディスク領域が必要です。
  - システムに Linux ユーティリティ `mount`、`rm`、`cp`、`umount`、`zcat`、`chroot`、`mkisofs` があることを確認します。これらのユーティリティはスクリプトによって使用されます。これらすべての Linux コマンドを実行する権限があることを確認します。
  - システムのカーネル バージョンは、Cisco ISO のカーネル バージョンより後の 3.16 以降である必要があります。

- Linux カーネルでサポートされている `libyaml rpm` が、ツールで `yaml` を正常に実行できることを確認します。
- ユーザは rpm リポジトリのセキュリティ rpm（k9sec-rpm）に対する適切な権限を持っている必要があります。それ以外の場合は、ゴールデン ISO の作成でセキュリティ rpm が無視されます。

- gisobuild スクリプトが実行されるシステムには、root クレデンシャルを使用する必要があります。

手順

ステップ 1 GISO を作成するオフライン システムまたは外部サーバに github の場所（https://github.com/ios-xr/gisobuild）からスクリプト `gisobuild.py` をコピーします。このシステムが上記の「はじめる前に」セクションに記載された前提条件を満たしていることを確認します。

ステップ 2 スクリプト `gisobuild.py` を実行し、ルータからゴールデン ISO を作成するためのパラメータを指定します。すべての RPM と SMU が同じディレクトリ内に存在することを確認します。ゴールデン ISO の作成に使用できる RPM と SMU の数は 128 です。

（注） `-i` オプションは必須で、`-r` と `-c` のいずれかまたはその両方を指定する必要があります。

```
[directory-path]$ gisobuild.py [-h] [-i <mini-x.iso>] [-r <rpm repository>]
[-c <config-file>] [-l <giso label>] [-m] [-v]
```

次に、スクリプトの出力例を示します。

値は次のとおりです。

- -i は mini-x.iso へのパスです
- -r は RPM リポジトリへのパスです
- -c は XR config ファイルへのパスです
- -l はゴールデン ISO ラベルです
- -h はヘルプ メッセージを表示します
- -v は、作成ツール `gisobuild.py` のバージョンです
- -m は、IOS XR から IOS XR 64 ビットに移行するための移行 tar を構築します

GISOは、指定されたディレクトリ内の各フォルダに配置されたRPMを使用して作成され、ログファイル `giso_summary.txt` および `gisobuild.log-<タイムスタンプ>` も含まれています。XR コンフィギュレーション ファイルはディレクトリ内に `router.cfg` として格納されます。

（注） GISO スクリプトは XR 設定の検証をサポートしていません。

次のタスク

ゴールデン ISO をルータにインストールします。

ゴールデン ISO のインストール

ゴールデン ISO（GISO）は、次のアクションを自動的に実行します。

- ホストおよびシステム管理 RPM をインストールします。
- RP でリポジトリと TFTP ブートをパーティションに分割します。
- システム管理モードおよび XR モードでソフトウェア プロファイルを作成します。
- XR RPM をインストールします。**show instal active** コマンドを使用して RPM のリストを表示します。
- XR 設定を適用します。XR モードで **show running-config** コマンドを使用して確認します。

手順

ステップ 1 次のいずれかのオプションを使用して、ルータに GISO イメージをダウンロードします。

- **PXE ブート**：ルータが起動すると、ブート モードが識別されます。PXE をブート モードとして検出すると、利用可能なすべてのイーサネット インターフェイスが起動し、各インターフェイスで DHClient が実行されます。DHClient スクリプトは HTTP または TFTP プロトコルを解析し、GISO がボックスにダウンロードされます。
- **USB ブートまたはディスク ブート**：ブート中に USB モードが検出され、GISO が識別されると、追加の RPM および XR 設定ファイルが抽出されてインストールされます。
- システムのアップグレード時の**システム アップグレード**では、**install add**、**install activate**、または **install update** コマンドを使用して GISO をインストールできます。
  - **非 GISO（GISO をサポートしていないイメージ）から GISO イメージへのシステム アップグレード**：システムが GISO をサポートしていないイメージを使用してバージョン 1 を実行している場合、システムは GISO をサポートするイメージのバージョン 2 に直接アップグレードすることはできません。その代わり、バージョン 1 をバージョン 2 ミニ ISO にアップグレードし、次にバージョン 2 GISO にアップグレードする必要があります。
  - **バージョン 1 GISO からバージョン 2 GISO へのリリースでのシステム アップグレード**：両方の GISO イメージの基本バージョンは同じでラベルが異なる場合、**install add** および **install activate** コマンドは同じバージョンの 2 つのイメージをサポートしませ

ん。その代わりに、**install update** コマンドを使用してデルタ RPM のみをインストールします。システムのリロードはデルタ RPM の再起動タイプに基づいています。

- **バージョン 1 GISO からバージョン 2 GISO へのリリース間でのシステム アップグレード**：両方の GISO イメージの基本バージョンが異なります。**install add** および **install activate** コマンド、または **install update** コマンドを使用して、システム アップグレードを実行します。ルータは、バージョン 2 GISO イメージを使用したアップグレード後にリロードされます。

ステップ 2 システム管理モードで **show install repository all** コマンドを実行し、ホスト、システム管理、および XR の RPM と基本 ISO を表示します。

ステップ 3 **show install package <golden-iso>** コマンドを実行し、RPM のリストおよび GISO に組み込まれているパッケージを表示します。

GISO 内の ISO、SMU、およびパッケージがルータにインストールされます。

ゴールデン ISO への置換のインストール

ゴールデン ISO（GISO）は、単一の操作でソフトウェア メンテナンス アップデート（SMU）の事前定義されたリストを持つバージョンにルータをアップグレードします。ただし、異なる SMU セットを使用した同じバージョンに更新するには、2 段階のプロセスが必要です。このプロセスでは、GISO をアップグレードしてデルタ SMU を追加し、使用されていない SMU を手動で非アクティブ化する必要があります。

この 2 段階のプロセスを回避するには、 コマンドを使用して、現在アクティブなバージョンを、新しく追加した GISO のイメージを含む完全なパッケージに置き換えます。

（注） **install updatereplace** キーワードは GISO でのみサポートされています。.mini および .rpm パッケージでは直接サポートされていません。

手順

ステップ 1 .

例：

```
RP/0/RP0/CPU0:ios#install  harddisk:/misc/disk1/<giso-image>.iso
 noprompt
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Install operation 11 started by root:
exec-timeout is suspended.
No install operation in progress at this moment
Label = More_Pkgs
ISO <giso-iso-image>.iso in input package list. Going to upgrade the system to
```

```
version <new-giso-image>.
System is in committed state
Current full-label: <giso-image>_R_Commit
Current only-label: R_Commit
Current label: R_Commit
Updating contents of golden ISO
Scheme : localdisk
Hostname : localhost
Username : None
SourceDir : /ws
Collecting software state..
Getting platform
Getting supported architecture
Getting active packages from XR
Getting inactive packages from XR
Getting list of RPMs in local repo
Getting list of provides of all active packages
Getting provides of each rpm in repo
Getting requires of each rpm in repo
Fetching .... <giso-image>.iso
Label within GISO: More_Pkgs
Skipping <platform>-mgbl-3.0.0.0-<release>.x86_64.rpm from GISO as it's active
Adding packages
        <platform>-golden-x-<release>-<Label>.iso
RP/0/RP0/CPU0:Jun 20 14:43:59.349 UTC: sdr_instmgr[1164]:
%INSTALL-INSTMGR-2-OPERATION_SUCCESS :

Install operation 12 finished successfully
Install add operation successful
Activating <platform>-golden-x-<release>-<Label>
Jun 20 14:44:05 Install operation 13 started by root:
  install activate pkg <platform>-golden-x-<release>-<Label> replace noprompt
Jun 20 14:44:05 Package list:
Jun 20 14:44:05     <platform>-golden-x-<release>-<Label>.iso
Jun 20 14:44:29 Install operation will continue in the background
exec-timeout is resumed.
RP/0/RP0/CPU0:ios#Jun 20 14:51:01 Install operation 13 finished successfully
RP/0/RP0/CPU0:Jun 20 14:51:01.416 UTC: sdr_instmgr[1164]:
%INSTALL-INSTMGR-2-OPERATION_SUCCESS :

Install operation 13 finished successfully
RP/0/RP0/CPU0:Jun 20 14:51:01.417 UTC: sdr_instmgr[1164]:
%INSTALL-INSTMGR-2-SYSTEM_RELOAD_INFO :
```

新しく追加された GISO のバージョンおよびラベルは、現在アクティブなバージョンのバージョンおよびラベルと比較されます。不一致が特定されると、新しいパーティションが作成され、完全なパッケージがインストールされます。インストール後、システムは新しく追加された GISO からイメージおよびパッケージをリロードします。

（注） 有効なラベルを持つシステムでアクティブ化または非アクティブ化すると、ラベルが無効になります。この操作は元に戻せません。たとえば、システムで **show version** コマンドを実行すると、ラベル `6.3.3.15I_633rev1005` が表示されます。システムで SMU がアクティブ化または非アクティブ化になると、ラベル `633rev1005` は無効になり、`show version` コマンドはラベルとして `6.3.3.15I` のみを表示します。

ステップ 2 show version

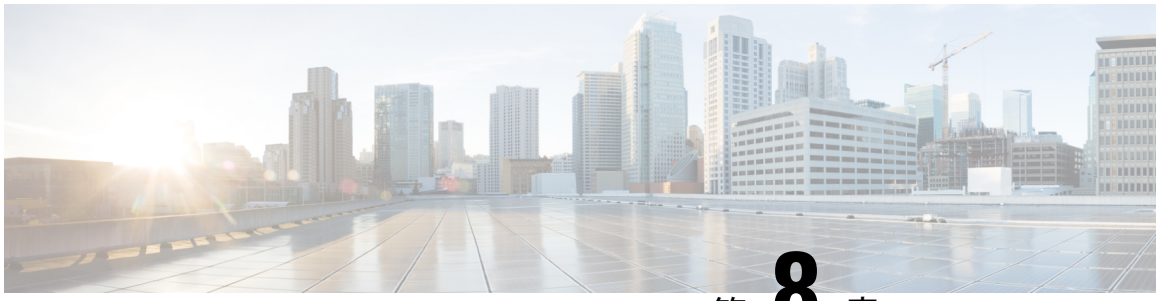
例：

```
RP/0/RP0/CPU0:ios#show version
Wed Jun 20 15:06:37.915 UTC
Cisco IOS XR Software, Version <new-giso-image>
Copyright (c) 2013-2018 by Cisco Systems, Inc.

Build Information:
Built By     : <user>
Built On     : <date>
Build Host   : <host-name>
Workspace    : <workspace-name>
Version      : <version>
Location    : <path>
Label        : <label-name>

cisco <platform> () processor
System uptime is 3 hours 51 minutes
```

システムは新しく追加された GISO からイメージおよびパッケージをリロードします。



第 8 章

ディザスタ リカバリ

この章で説明する内容は次のとおりです。

USB ドライブを使用した起動

ブート可能な USB ドライブを使用して、システム アップグレードの目的でルータのイメージを再適用したり、起動に失敗した場合にルータを起動したりします。ブート可能な USB ドライブは圧縮ブート ファイルを使用して作成できます。

圧縮ブート ファイルを使用したブート可能な USB ドライブの作成

圧縮ブート ファイルを USB ドライブにコピーすると、ブート可能な USB ドライブが作成されます。圧縮ファイルの内容が展開されると、USB ドライブがブート可能になります。

（注） USB ドライブからの読み込みまたはブートに失敗した場合は、ドライブが正しく挿入されていることを確認してください。ドライブが正しく挿入されていても USB ドライブから読み込めない場合は、別のシステムで USB の内容を確認してください。

このタスクは、ローカル マシンで利用できる Windows、Linux、または MAC オペレーティング システムを使用して実行できます。ここで説明する一般的な手順をそれぞれ実行するための操作は、使用中のオペレーティング システムによって異なります。

始める前に

- ストレージ容量が 8 GB（最小）~32 GB（最大）の USB ドライブにアクセスできるようにします。USB 2.0 および USB 3.0 がサポートされています。

- 圧縮ブート ファイルを cisco.com のソフトウェア ダウンロード ページからローカル マシンにコピーします。圧縮ブート ファイルのファイル名の形式は、*ncs560-usb-boot-<release_number_zip>* です。

手順

ステップ 1 USB ドライブをローカル マシンに接続し、Windows オペレーティング システムまたは Apple MAC ディスク ユーティリティを使用して FAT32 または MS-DOS ファイル システムでフォーマットします。

ステップ 2 圧縮ブート ファイルを USB ドライブにコピーします。

ステップ 3 コピー処理が正常に行われたことを確認します。確認するには、コピー元とコピー先でファイル サイズを比較します。さらに、MD5 チェックサム値を確認します。

ステップ 4 圧縮ブート ファイルを USB ドライブ内で解凍して内容を展開します。これにより、USB ドライブがブート可能なドライブに変換されます。

（注） 圧縮ファイルの内容（「EFI」および「boot」ディレクトリ）は、USB ドライブのルートに直接展開する必要があります。解凍アプリケーションによって展開ファイルが新しいフォルダに配置された場合は、「EFI」および「boot」ディレクトリを USB ドライブのルートに移動してください。

ステップ 5 ローカル マシンから USB ドライブを取り出します。

次のタスク

ブート可能な USB ドライブを使用して、ルータの起動またはイメージのアップグレードを実行します。

iPXE を使用した起動

iPXE は、管理インターフェイスのネットワーク カードに含まれ、ルータのシステム ファームウェア（UEFI）レベルで動作するプリブート実行環境です。iPXE は、システムを再イメージするために使用され、ブートに失敗した場合や有効なブート可能なパーティションがない場合にルータを起動します。iPXE は ISO イメージをダウンロードして、イメージのインストールを進行させ、最後に新しいインストール内でブートストラップを行います。

iPXE はブート ローダとして機能し、システムを起動するイメージをプラットフォーム ID（PID）、シリアル番号、または管理 MAC アドレスに基づいて柔軟に選択できるようにします。iPXE は DHCP サーバのコンフィギュレーション ファイルで定義する必要があります。

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（ZTP）は、iPXEを使用してルータでソフトウェアをインストールした後の自動プロビジョニングに役立ちます。

ZTP の自動プロビジョニングでは以下の手順を実行します。

- **設定**：コンフィギュレーション ファイルをダウンロードおよび実行します。ZTP でコンフィギュレーションとして処理されるように、ファイルの最初の行に `!! IOS XR` が含まれている必要があります。
- **スクリプト**：スクリプト ファイルをダウンロードおよび実行します。スクリプト ファイルには、タスクを完了するためのプログラムによるアプローチが含まれています。たとえばIOS XR コマンドを使用して作成されたスクリプトは、パッチ アップグレードを実行します。ZTP でコンフィギュレーションとして処理されるように、ファイルの最初の行に `#! /bin/bash` または `#! /bin/sh` が含まれている必要があります。

DHCP サーバの設定

DHCP サーバは、IPv4 か IPv6、またはその両方の通信プロトコルに対して設定する必要があります。次に、Linux システムで実行されている ISC-DHCP サーバの例を示します。

始める前に

- ネットワーク管理者またはシステムの設計担当者に問い合わせて、管理インターフェイスの IP アドレスおよびサブネット マスクを入手します。
- RP の物理ポート イーサネット 0 は管理ポートです。ポートが管理ネットワークに接続されていることを確認します。
- サーバが DHCP パケットを処理できるようにファイアウォールを有効にします。
- DHCPv6 の場合、IPv6 アドレスの取得方法を示すルーティング アドバタイズメント（RA）メッセージをネットワーク内のすべてのノードに送信する必要があります。クライアントが DHCP 要求を送信できるようにルータ アドバタイズ デーモン（radvd。yum install radvd を使用してインストールします）を設定します。次に例を示します。

```
interface eth3
{
        AdvSendAdvert on;
        MinRtrAdvInterval 60;
        MaxRtrAdvInterval 180;
        AdvManagedFlag on;
        AdvOtherConfigFlag on;
        prefix 2001:1851:c622:1::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr off;
        };
};
```

- HTTP サーバは DHCP サーバと同じサーバにも、別のサーバにも設定できます。IP アドレスが DHCP サーバから割り当てられた後、ルータは HTTP サーバに接続してイメージをダウンロードします。

（注） ゼロ タッチ プロビジョニング（ZTP）は、Cisco IOS XR リリース 6.6. x の Cisco NCS 560 ルータではサポートされていません。

手順

ステップ 1 `dhcpd.conf` ファイル（IPv4、IPv6、または両方の通信プロトコル用）、`dhcpv6.conf` ファイル（IPv6 用）、またはその両方のファイルを `/etc/` または `/etc/dhcp` ディレクトリに作成します。このコンフィギュレーション ファイルには、スクリプトへのパス、ISO インストール ファイルの場所、プロビジョニング設定ファイルの場所、ルータのシリアル番号、MAC アドレスなどのネットワーク情報が保存されます。

ステップ 2 DHCP サーバが稼働したら、サーバをテストします。たとえば、IPv4 の場合は次のようになります。

- ルータの MAC アドレスを使用した場合：

  （注） `host` ステートメントを使用すると、DNS に使用される固定アドレスが提供されますが、要求内でオプション 77 が iPXE に設定されていることを確認します。このオプションは、必要に応じてブートファイルをシステムに提供するために使用されます。

  上記の設定が正常に行われていることを確認します。
- ルータのシリアル番号を使用した場合：ルータのシリアル番号は BIOS から取得され、ID として使用されます。

ステップ 3 DHCP を再起動します。

```
killall dhcpd
/usr/sbin/dhcpd -f -q -4 -pf /run/dhcp-server/dhcpd.pid
-cf /etc/dhcp/dhcpd.conf ztp-mgmt &
```

例

次に、`dhcpd.conf` ファイルの例を示します。

```
allow bootp;
allow booting;
ddns-update-style interim;
option domain-name "cisco.com";
option time-offset -8;
ignore client-updates;
default-lease-time 21600;
```

```
max-lease-time 43200;
option domain-name-servers <ip-address-server1>, <ip-address-server2>;
log-facility local0;
 :
subnet <subnet> netmask <netmask> {
  option routers <ip-address>;
  option subnet-mask <subnet-mask>;
  next-server <server-addr>;
}
  :
host <hostname> {
  hardware ethernet e4:c7:22:be:10:ba;
  fixed-address <address>;
  filename "http://<address>/<path>/<image.bin>";
}
```

次に、dhcpd6.conf ファイルの例を示します。

```
option dhcp6.name-servers <ip-address-server>;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/db/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
subnet6 <subnet> netmask <netmask> {
       range6 2001:1851:c622:1::2 2001:1851:c622:1::9;
        option dhcp6.bootfile-url "http://<address>/<path>/<image.bin>";
```

次のタスク

ZTP を呼び出します。

ZTP の呼び出し

ZTP は XR 名前空間内と、管理インターフェイスおよびラインカードインターフェイスの場合はグローバル VPN ルーティング/転送（VRF）名前空間内で実行されます。

始める前に

DHCP サーバが設定されていることを確認します。詳細については、DHCP サーバの設定 （61 ページ）を参照してください。

手順

dhcpd.conf ファイルを編集して、ZTP の機能を利用します。

次に、iPXE と ZTP を含む DHCP サーバの設定例を示します。

```
host <host-name>
{
hardware ethernet <router-serial-number or mac-id>;
fixed-address <ip-address>;
  if exists user-class and option user-class = "iPXE" {
  # Image request, so provide ISO image
  filename "http://<ip-address>/<directory>/";
```

```
  } else
{
  # Auto-provision request, so provide ZTP script or configuration
  filename "http://<ip-address>/<script-directory-path>/";
  #filename "http://<ip-address>/<script-directory-path>/
  }
}
```

（注） 自動プロビジョニング用に一度に提供できるのは、ZTP .script ファイルまたは .cfg ファイルのいずれかのみです。

この設定では、インストール時に を使用してシステムを起動し、その後 XR VM が起動した時点で をダウンロードして実行します。

手動による ZTP の呼び出し

ZTP は、変更されたワンタッチ プロビジョニング手法を使用して手動で呼び出すこともできます。このプロセスでは、次の手順を実行する必要があります。

始める前に

設定ファイルを使用して、XR で起動され、DHCP が呼び出されるインターフェイスのリストを指定することができます。/pkg/etc/ztp.config はプラットフォーム固有のファイルで、追加のインターフェイスを使用するかどうかをプラットフォームが指定できます。

```
#
# List all the interfaces that ZTP will consider running on. ZTP will attempt
# to bring these interfaces. At which point dhclient will be able to use them.
#
# Platforms may add dynamically to this list.
#
#ZTP_DHCLIENT_INTERFACES=" \
#    Gi0_0_0_0 \
#"
...
```

手順

ステップ 1 ルータを起動します。

ステップ 2 手動でログインします。

ステップ 3 インターフェイスを有効にします。

ステップ 4 **ztp initiate** コマンドを使用して新しい ZTP DHCP セッションを手動で呼び出します。

```
Router#ztp initiate
```

たとえば、GigabitEthernet インターフェイス 0/0/0/0 で DHCP 要求を送信するには、次のコマンドを実行します。

```
Router#ztp initiate debug verbose interface GigabitEthernet0/0/0/0
```

プラットフォームが別の方法で設定していない限り、ZTP はデフォルトで管理ポート上で実行されます。ログは /disk0:/ztp/ztp/log の場所に記録されます。

（注） 40G インターフェイスを 4 つの個別の 10G インターフェイスに設定するには、**ztp breakout nosignal-stay-in-breakout-mode** コマンドを使用します。

（注） データポートブレークアウトを有効にし、検出されたすべてのデータポートインターフェイスおよびラインカードインターフェイスで DHCP セッションを呼び出すには、**ztp breakout** コマンドを使用します。

```
Router#ztp breakout debug verbose
Router#ztp initiate dataport debug verbose
Invoke ZTP?(this may change your configuration) [confirm] [y/n]:
```

プロンプトを上書きするには：

```
Router#ztp initiate noprompt
Invoke ZTP?(this may change your configuration) [confirm] [y/n]:

ZTP will now run in the background.
Please use "show logging" or look at /disk0:/ztp/ztp/log to check progress.
```

ZTP はデフォルトで起動している管理インターフェイス上で動作します。

ステップ 5 ZTP セッションを終了するには、**ztp terminate** コマンドを使用します。

次のタスク

iPXE を使用してルータを起動します。

iPXE を使用したルータの起動

iPXE ブートを使用する前に、次のことを確認してください。

- DHCP サーバが設定され、動作している。
- **admin** コマンドを使用してシステム管理コンソールにログインしている。

ルータのイメージを再作成するために、次のコマンドを実行して iPXE ブート プロセスを呼び出します。

```
hw-module location all bootmedia network reload
```

例：

```
sysadmin-vm:0_RP0# hw-module location all bootmedia network reload
Wed Dec 23 15:29:57.376 UTC
Reload hardware module ? [no,yes]
```

次の例は、コマンドの出力を示しています。

```
iPXE 1.0.0+ (3e573) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
```

```
Trying net0...
net0: c4:72:95:a6:14:e1 using dh8900cc on PCI01:00.1 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 c4:72:95:a6:14:e1)..................... Ok << Talking to DHCP/PXE server
 to obtain network information
net0: 10.37.1.101/255.255.0.0 gw 10.37.1.0
net0: fe80::c672:95ff:fea6:14e1/64
net0: 2001:1800:5000:1:c672:95ff:fea6:14e1/64 gw fe80::20c:29ff:fefb:b9fe
net1: fe80::c672:95ff:fea6:14e3/64 (inaccessible)
Next server: 10.37.1.235
Filename: http://10.37.1.235/


http://10.37.1.235/ ... 58% << Downloading file as indicated by DHCP/PXE server to boot
 install image
```