



NETCONF プロトコル

- [NETCONF プロトコルの制約事項 \(1 ページ\)](#)
- [NETCONF プロトコルの概要 \(1 ページ\)](#)
- [NETCONF プロトコルの設定方法 \(8 ページ\)](#)
- [NETCONF プロトコルのコンフィギュレーションの確認 \(12 ページ\)](#)
- [NETCONF プロトコルの関連資料 \(14 ページ\)](#)
- [NETCONF プロトコルの機能情報 \(15 ページ\)](#)

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 は非常に独自のであり、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、IOSXE、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 はシンプルなり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>
```

```

        </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。

候補コンフィギュレーション サポート

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。

候補データストアは、デバイスの実行コンフィギュレーションのコピーを保存する一時的な作業領域となります。実行コンフィギュレーションをデバイスにコミットする前に、実行コンフィギュレーションを作成して変更することができます。候補機能は、NETCONF 機能 `urn:ietf:params:netconf:capability:candidate:1.0` により示されます。この NETCONF 機能は、デバイスが候補データストアをサポートしていることを示します。ユーザはこの共有データストアを使用して、デバイスの実行コンフィギュレーションに影響を与えることなく、デバイスのコンフィギュレーションを作成、追加、削除、変更できます。コミット操作では、デバイスのコンフィギュレーションが「候補」から「実行」のコンフィギュレーションにプッシュされます。「候補」データストアが有効になっていると、「実行」のデータストアには NETCONF セッションを介して書き込むことができず、すべてのコンフィギュレーションは候補を通じてのみコミットされます。つまり、稼働中の設定を直接変更できる NETCONF 機能は、候補では有効になりません。



- (注) 候補データストアは共有データストアであることに留意してください。複数の NETCONF セッションが内容を同時に変更する可能性があります。したがって、内容を変更する前にデータストアをロックして、コミットが競合しないようにし、最終的にコンフィギュレーションの変更が失われる可能性を防ぐことが重要になります。

候補の NETCONF 操作

候補データストアでは次の操作を実行できます。



- (注) この項の情報は RFC 6241 の 8.3.4 項を参考にしています。詳細と正確な RPC については、RFC を参照してください。

ロック

<lock> RPC は、ターゲットのデータストアをロックするために使用します。これにより、他のユーザはロックされたデータストアのコンフィギュレーションを変更できなくなります。ロック操作では候補データと実行データの両方をロックできます。



- (注) 候補データストアのロックは、Cisco IOS のコンフィギュレーションのロックや実行コンフィギュレーションのロックに影響を与えません。逆も同様です。

コミット

<commit> RPC は、候補コンフィギュレーションをデバイスの実行コンフィギュレーションにコピーします。「コミット」操作は、候補コンフィギュレーションを更新してコンフィギュレーションをデバイスにプッシュした後に実行する必要があります。

実行または候補のデータストアのいずれかが別の NETCONF セッションによってロックされている場合、<commit> RPC は RPC エラー応答で失敗します。<error-tag> は <in-use> となり、<error-info> にはロックを保持している NETCONF セッションのセッション ID が示されます。conf t lock モードに移行してグローバルロックを使用し、「実行」コンフィギュレーションをロックすることもできますが、コミット操作は RPC エラー応答で失敗し、error-tag の値は <in-use>、セッション ID は「0」になります。

コンフィギュレーション編集

候補コンフィギュレーションは、コンフィギュレーションを変更するための edit-config (コンフィギュレーション編集) 操作のターゲットとして使用できます。デバイスの実行コンフィギュレーションに影響を与えることなく、候補コンフィギュレーションを変更できます。

廃棄

候補コンフィギュレーションに加えられた変更を削除するには、discard（廃棄）操作を実行して候補コンフィギュレーションを実行コンフィギュレーションに戻します。

たとえば、NETCONFセッションAによって候補データストアの内容が変更されている場合、セッションBが候補データストアをロックしようとするするとロックは失敗します。NETCONFセッションBでは候補をロックする前に、他のNETCONFセッションから候補データストアの未解決のコンフィギュレーションの変更を削除するために<discard>操作を実行する必要があります。

ロック解除

ロック、edit-config（コンフィギュレーション編集）、コミットなどで候補コンフィギュレーションを操作した後、ロック解除RPCでターゲットとしてcandidateを指定することによって、データストアをロック解除できます。これで、他のセッションでのすべての操作に候補データストアを使用できるようになります。

候補データストアに対する未解決の変更で不具合が発生した場合、コンフィギュレーションの回復が困難になり、他のセッションで問題が生じる可能性があります。問題を回避するため、未解決の変更は、「NETCONFセッションの障害」で暗黙的にロックが解除されたとき、またはロック解除操作を使用して明示的にロックが解除されたときに廃棄される必要があります。

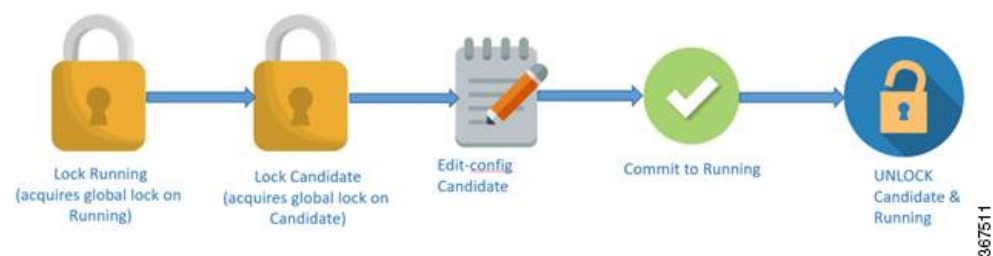
コンフィギュレーション取得、コンフィギュレーションコピー、コンフィギュレーション検証

候補データストアは、get-config（コンフィギュレーション取得）、copy-config（コンフィギュレーションコピー）、またはvalidate（コンフィギュレーション検証）のどの操作でも、ソースまたはターゲットとして使用できます。候補データストアの変更をデバイスにコミットせずに、コンフィギュレーションの検証のみを行う場合は、<validate>RPCの後にdiscardの操作を付けることで使用できます。

候補データストアの変更

次の図は、候補データストアを介してデバイスコンフィギュレーションを変更する場合に推奨されるベストプラクティスを示しています。

図 1: 候補データストアの変更手順



1. 実行データストアをロックします。
2. 候補データストアをロックします。

3. edit-config RPC とターゲットの候補を使用して、候補コンフィギュレーションを変更します。
4. 候補コンフィギュレーションを、実行コンフィギュレーションにコミットします。
5. 候補データストアと実行データストアをロック解除します。

候補サポートの設定

候補データストア機能は、**netconf-yang feature candidate-datastore** コマンドを使用して有効にすることができます。データストアの状態が「実行」から「候補」、またはその逆に変わると、変更を有効にするために **netconf-yang** または **restconf** の再起動が行われることをユーザーに通知する警告メッセージが表示されます。候補が有効になっていると、実行のデータストアには NETCONF セッションを介して書き込むことができず、すべてのコンフィギュレーションは候補を通じてのみコミットされます。つまり、稼働中の設定を直接変更できる Netconf 機能は、候補では有効になりません。

netconf-yang または **restconf confd** プロセスの開始時に候補または実行のデータストアの選択がコンフィギュレーションで指定されている場合は、次のような警告が表示されます。

```
Device(config)# netconf-yang feature candidate-datastore  
netconf-yang initialization in progress - datastore transition not allowed, please try  
again after 30 seconds
```

netconf-yang または **restconf confd** プロセスの開始後に候補または実行の選択が行われた場合は、次のように適用されます。

- **netconf-yang feature candidate-datastore** コマンドが設定されている場合は、コマンドによって「候補」データストアが有効になり、次の警告が出力されます。

```
"netconf-yang and/or restconf is transitioning from running to candidate netconf-yang  
and/or restconf will now be restarted,  
and any sessions in progress will be terminated".
```

- **netconf-yang feature candidate-datastore** コマンドが削除された場合は、コマンドによって「候補」データストアが無効になり、「実行」データストアが有効になり、次の警告が出力されます。

```
netconf-yang and/or restconf is transitioning from candidate to running netconf-yang  
and/or restconf will now be restarted,  
and any sessions in progress will be terminated".
```

- **netconf-yang** または **restconf** が再起動すると、進行中のセッションは失われます。



(注) 候補データストアは共有データストアです。つまり、複数の NETCONF セッションで内容を同時に変更できます。したがって、内容を変更する前にユーザがデータストアをロックして、コミットが競合しないようにし、最終的にコンフィギュレーションの変更が失われる（つまり、他のユーザによってコンフィギュレーションが変更され、コミットが発行されて、コンフィギュレーションが上書きされる）可能性を防ぐことが重要になります。

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、『[Public Key Infrastructure Configuration Guide, Cisco IOS XE Gibraltar 16.10.x](#)』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。

手順の概要

1. **enable**
2. **configure terminal**
3. **username name privilege level password password**
4. **aaa new-model**
5. **aaa authentication login default local**
6. **aaa authorization exec default local**
7. **end**

手順の詳細

	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	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。 <ul style="list-style-type: none"> • privilege level : ユーザの権限レベルを設定します。NETCONF プロトコルの場合は、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 new-model 例：	(任意) 許可、認証、アカウントिंग (AAA) を有効にします。

	コマンドまたはアクション	目的
	Device(config)# aaa new-model	aaa new-model コマンドを設定する場合は、AAA 認証および許可が必要です。
ステップ 5	aaa authentication login default local 例： Device(config)# aaa authentication login default local	ローカルユーザ名データベースを使用するログイン認証を設定します。 (注) NETCONF プロトコルでは、デフォルトの AAA 認証ログイン方式のみがサポートされます。 <ul style="list-style-type: none"> 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。 default キーワードにより、ローカルユーザデータベース認証がすべてのポートに適用されます。
ステップ 6	aaa authorization exec default local 例： Device(config)# aaa authorization exec default local	ユーザの AAA 許可を設定し、ローカルデータベースを確認して、そのユーザに EXEC シェルの実行を許可します。 <ul style="list-style-type: none"> 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。 default キーワードにより、ローカルユーザデータベース認証がすべてのポートに適用されます。
ステップ 7	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。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **netconf-yang**
4. **netconf-yang feature candidate-datastore**
5. **exit**

手順の詳細

	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	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。

手順の概要

1. IOS で SNMP 機能を有効にします。

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。
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。

手順の詳細

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。

手順の概要

1. **show netconf-yang datastores**
2. **show netconf-yang sessions**
3. **show netconf-yang sessions detail**
4. **show netconf-yang statistics**
5. **show platform software yang-management process**

手順の詳細

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 1 : `show platform software yang-management process` のフィールドの説明

フィールド	説明
<code>confd</code>	コンフィギュレーションデーモン
<code>nesd</code>	ネットワーク要素シンクロナイザデーモン
<code>syncfd</code>	デーモンからの同期
<code>ncsshd</code>	NETCONF セキュア シェル (SSH) デーモン
<code>dmiauthd</code>	デバイス管理インターフェイス (DMI) 認証デーモン
<code>vtyserverutild</code>	VTY サーバユーティリティデーモン
<code>opdatamgrd</code>	運用データ マネージャ デーモン
<code>nginx</code>	NGINX Web サーバ
<code>ndbmand</code>	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	<i>YANG : Network Configuration Protocol (NETCONF) 向けデータモデリング言語</i>
RFC 6241	ネットワーク設定プロトコル (<i>NETCONF</i>)
RFC 6536	ネットワーク設定プロトコル (<i>NETCONF</i>) アクセス制御モデル
RFC 8040	<i>RESTCONF</i> プロトコル

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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。</p> <p>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。</p>	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 2: NETCONF プロトコルの機能情報

機能名	リリース	機能情報
NETCONF プロトコル	Cisco IOS XE Denali 16.3.1	<p>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。</p> <p>次のコマンドが導入されました： netconf-yang</p> <p>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。</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ
	Cisco IOS XE Everest 16.5.1a	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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 IOS XE Fuji 16.8.1a	

機能名	リリース	機能情報
		<p>Cisco IOS XE Fuji 16.8.1a 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。</p> <ul style="list-style-type: none">• Cisco 1000 シリーズ サービス統合型ルータ• Cisco ASR 900 シリーズ アグリゲーション サービス ルータ• Cisco ASR 920 シリーズ アグリゲーション サービス ルータ• Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ• Cisco CBR-8 シリーズ ルータ• Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Fuji 16.9.2	<p>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。</p> <ul style="list-style-type: none">• Cisco Catalyst 9200 および 9200L シリーズ スイッチ• Cisco Catalyst 9300L SKU

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	<p>NETCONF および RESTCONF プロトコルの IPv6 のサポート。この機能は、次のプラットフォームに実装されています。</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ • 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

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	<p>NETCONF プロトコルは、グローバルロックおよび応答しなくなったセッションを kill する機能をサポートしています。この機能は、次のプラットフォームに実装されています。</p> <ul style="list-style-type: none">• Cisco 1100 シリーズ サービス統合型ルータ• Cisco 4000 シリーズ サービス統合型ルータ• 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

機能名	リリース	機能情報
NETCONF : 候補コンフィギュレーションサポート	Cisco IOS XE Fuji 16.9.1	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ • 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 <p>次のコマンドが導入されました： netconf-yang feature candidate-datastore</p>

翻訳について

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