ネットワーク設定プロトコル

ネットワーク設定プロトコル（NETCONF）は、ネットワーク デバイスの管理、設定データの取得、および新しい設定データのアップロードと操作を行うための簡単なメカニズムを定義するものです。 NETCONF では、設定データおよびプロトコル メッセージとして拡張可能マークアップ言語（XML）ベースのデータ符号化を使用します。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の警告および機能情報については、『Bug Search Tool』およびご使用のプラットフォームとソフトウェア リリースに対応したリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

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。

NETCONF の前提条件

netconf max-session コマンドで指定されているように、各 NETCONF セッションで vty 行が必要です。

NETCONF の概要

NETCONF 通知

NETCONF は、NETCONF 上で設定変更の通知を送信します。 通知は、設定変更が行われたことを示すイベントです。 変更には、設定の追加、削除、または修正があります。 通知は、適切に行われた設定作業の最後に、設定内で変更された設定の各行について個別のメッセージではなく、一連の変更を示す 1 つのメッセージとして送信されます。

NETCONF の設定方法

NETCONF ネットワーク マネージャ アプリケーションの設定

手順の概要

1. NETCONF を SSH サブシステムとして呼び出すように、NETCONF ネットワーク マネージャ アプリケーションを設定するには、次の CLI 文字列を使用します。
2. NETCONF セッションの確立後すぐに、<hello> を含む次のような XML 文書を送信することによって、サーバの機能を示します。
3. 次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションが NETCONF 通知を送受信できるようにします。
4. NETCONF ネットワーク マネージャ アプリケーションの NETCONF 通知の送信または受信を停止するには、次の XML 文字列を使用します。

手順の詳細

ステップ 1 NETCONF を SSH サブシステムとして呼び出すように、NETCONF ネットワーク マネージャ アプリケーションを設定するには、次の CLI 文字列を使用します。

例：

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

ステップ 2 NETCONF セッションの確立後すぐに、<hello> を含む次のような XML 文書を送信することによって、サーバの機能を示します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
    <hello>
      <capabilities>
        <capability>
            urn:ietf:params:xml:ns:netconf:base:1.0
          </capability>
          <capability>
            urn:ietf:params:ns:netconf:capability:startup:1.0
          </capability>
       </capabilities>
    <session-id>4<session-id>
</hello>]]>]]>
```

クライアントは、<hello> を含む XML 文書を送信して応答します。

例：

```
<?xml version="1.0" encoding="UTF-8"?>
 <hello>
   <capabilities>
       <capability>
           urn:ietf:params:xml:ns:netconf:base:1.0
     </capability>
    </capabilities>
</hello>]]>]]>
```

（注） この例では、サーバの <hello> メッセージの送信後にクライアントのメッセージが続くことになっていますが、NETCONF サブシステムの初期化後すぐに、両サイドからほぼ同時にメッセージが送信されます。

ヒント すべての NETCONF 要求は、要求の終わりを示す]]>]]> で終わる必要があります。]]>]]> のシーケンスが送信されるまで、デバイスは要求を処理しません。

特定の例については、「例：NETCONF over SSHv2 の設定」を参照してください。

ステップ 3 次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションが NETCONF 通知を送受信できるようにします。

例：

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>
```

ステップ 4 NETCONF ネットワーク マネージャ アプリケーションの NETCONF 通知の送信または受信を停止するには、次の XML 文字列を使用します。

例：

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>
```

NETCONF ペイロードの配信

NETCONF ペイロードをネットワーク マネージャ アプリケーションに配信するには、次の XML 文字列を使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.cisco.com/cpi_10/schema" elementFormDefault="qualified"
 attributeFormDefault="unqualified" xmlns="http://www.cisco.com/cpi_10/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <!--The following elements define the cisco extensions for the content of the filter
element in a <get-config> request. They allow the client to specify the format of the
response and to select subsets of the entire configuration to be included.-->
   <xs:element name="config-format-text-block">
      <xs:annotation>
         <xs:documentation>If this element appears in the filter, then the client is
requesting that the response data be sent in config command block format.</xs:documentation>

      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-text-cmd">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-xml">
      <xs:annotation>
         <xs:documentation>When this element appears in the filter of a get-config request,
 the results are to be returned in E-DI XML format. The content of this element is treated
 as a filter.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="xs:anyType"/>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
   <!--These elements are used in the filter of a <get> to specify operational data to
return.-->
   <xs:element name="oper-data-format-text-block">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="show" type="xs:string" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="oper-data-format-xml">
      <xs:complexType>
         <xs:sequence>
```

```
            <xs:any/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <!--When confing-format-text format is specified, the following describes the content
of the data element in the response-->
   <xs:element name="cli-config-data">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="cmd" type="xs:string" maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>Content is a command. May be multiple
lines.</xs:documentation>
               </xs:annotation>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="cli-config-data-block" type="xs:string">
      <xs:annotation>
         <xs:documentation>The content of this element is the device configuration as it
would be sent to a terminal session. It contains embedded newline characters that must be
preserved as they represent the boundaries between the individual command
lines</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="text-filter-spec">
      <xs:annotation>
         <xs:documentation>If this element is included in the config-format-text element,
then the content is treated as if the string was appended to the "show running-config"
command line.</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="cli-oper-data-block">
      <xs:complexType>
         <xs:annotation>
            <xs:documentation> This element is included in the response to get operation.
Content of this element is the operational data in text format.</xs:documentation>
         </xs:annotation>
         <xs:sequence>
            <xs:element name="item" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="show"/>
                     <xs:element name="response"/>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:schema>
```

NETCONF 通知のフォーマット

NETCONF ネットワーク マネージャ アプリケーションは、.xsd スキーマ ファイルを使用して、NETCONF ネットワーク マネージャ アプリケーションと NETCONF over SSHv2 または NETCONF over BEEP が稼働するデバイスとの間で送信される XML NETCONF 通知メッセージのフォーマットを記述します。 それらのファイルはブラウザまたはスキーマ読み取りツールで表示できます。これらのスキーマを使用して XML の妥当性を検証できます。 これらのスキーマで記述するのは、交換されるデータのフォーマットであって内容ではありません。

NETCONF は <edit-config> 機能を使用して、特定の設定すべてを特定のターゲット設定にロードします。 この新しい設定を入力した場合、ターゲット設定は置き換えられません。 ターゲット設定は、要求の送信元のデータおよび要求された動作に応じて変更されます。

次に、CLI、CLI ブロック、および XML の各フォーマットの NETCONF <edit-config> 機能のスキーマを示します。

NETCONF <edit-config> 要求：CLI フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cli-config-data>
<cmd>hostname test</cmd>
            <cmd>interface fastEthernet0/1</cmd>
            <cmd>ip address 192.168.1.1 255.255.255.0</cmd>
</cli-config-data>
      </config>
   </edit-config>
</rpc>]]>]]>
```

NETCONF <edit-config> 応答: CLI フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:netconf:base:1.0">
   <ok/>
</rpc-reply>]]>]]>
```

NETCONF <edit-config> 要求：CLI ブロック フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="netconf.mini.edit.3">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cli-config-data-block>
            hostname bob
            interface fastEthernet0/1
            ip address 192.168.1.1 255.255.255.0
         </cli-config-data-block>
      </config>
   </edit-config>
</rpc>]]>]]>
```

NETCONF <edit-config> 応答：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="netconf.mini.edit.3" xmlns="urn:ietf:params:netconf:base:1.0">
   <ok/>
</rpc-reply>]]>]]>
```

次に、CLI および CLI ブロックの各フォーマットの NETCONF <get-config> 機能のスキーマを示します。

NETCONF <get-config> 要求：CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-config>
      <source>
```

```
         <running/>
      </source>
      <filter>
         <config-format-text-cmd>
            <text-filter-spec> | inc interface </text-filter-spec>
         </config-format-text-cmd>
</filter>
   </get-config>
</rpc>]]>]]>
```

NETCONF <get-config> 応答: CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
      <cli-config-data>
         <cmd>interface FastEthernet0/1</cmd>
         <cmd>interface FastEthernet0/2</cmd>
      </cli-config-data>
   </data>
</rpc-reply>]]>]]>
```

NETCONF <get-config> 要求：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <config-format-text-block>
            <text-filter-spec> | inc interface </text-filter-spec>
         </config-format-text-block>
      </filter>
   </get-config>
</rpc>]]>]]>
```

NETCONF <get-config> 応答：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
      <cli-config-data-block>
         interface FastEthernet0/1
         interface FastEthernet0/2
      </cli-config-data-block>
   </data>
</rpc-reply>]]>]]>
```

NETCONF は <get> 機能を使用して、設定およびデバイスの状態情報を取得します。 NETCONF <get> フォーマットは、Cisco IOS **show** コマンドに相当します。 <filter> パラメータは、システム設定およびデバイス状態データの取得部分を指定します。 <filter>パラメータが空の場合は、何も返されません。

次に、CLI および CLI ブロックの各フォーマットの <get> 機能のスキーマを示します。

NETCONF <get> 要求：CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
       <filter>
```

```
          <config-format-text-cmd>
             <text-filter-spec> | include interface </text-filter-spec>
          </config-format-text-cmd>
          <oper-data-format-text-block>
             <exec>show interfaces</exec>
             <exec>show arp</exec>
          </oper-data-format-text-block>
       </filter>
    </get>
 </rpc>]]>]]>
```

NETCONF <get> 応答: CLI フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
       <cli-config-data>
<cmd>interface Loopback0</cmd>
<cmd>interface GigabitEthernet0/1</cmd>
<cmd>interface GigabitEthernet0/2</cmd>
</cli-config-data>
<cli-oper-data-block>
          <item>
             <exec>show interfaces</exec>
             <response>
                <!-- output of "show interfaces" ------>
             </response>
          </item>
          <item>
             <exec>show arp</exec>
             <response>
                <!-- output of "show arp" ------>
             </response>
          </item>
       </cli-oper-data-block>
    </data>
</rpc-reply>]]>]]>
```

NETCONF <get> 要求：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
       <filter>
          <config-format-text-block>
             <text-filter-spec> | include interface </text-filter-spec>
          </config-format-text-block>
          <oper-data-format-text-block>
             <exec>show interfaces</exec>
             <exec>show arp</exec>
          </oper-data-format-text-block>
       </filter>
    </get>
 </rpc>]]>]]>
```

NETCONF <get> 応答：CLI ブロック フォーマット

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
       <cli-config-data-block>
interface Loopback0
interface GigabitEthernet0/1
interface GigabitEthernet0/2
       </cli-config-data-block>
       <cli-oper-data-block>
          <item>
```

```
            <exec>show interfaces</exec>
            <response>
               <!-- output of "show interfaces" ------>
            </response>
         </item>
         <item>
            <exec>show arp</exec>
            <response>
               <!-- output of "show arp" ------>
            </response>
         </item>
      </cli-oper-data-block>
   </data>
</rpc-reply>]]>]]>
```

NETCONF セッションのモニタリングおよびメンテナンス

（注）
- 4 個以上の同時 NETCONF セッションを設定する必要があります。
- 最大 16 個の同時 NETCONF セッションを設定できます。
- NETCONF では SSHv1 はサポートされません。

手順の概要

1. **enable**
2. **show netconf** {**counters** | **session**| **schema**}
3. **debug netconf** {**all** | **error**}
4. **clear netconf** {**counters** | **sessions**}

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	**show netconf** {**counters**	**session**| **schema**} 例： `Device# show netconf counters`	NETCONF 情報を表示します。

	コマンドまたはアクション	目的
ステップ 3	**debug netconf** {**all** | **error**} 例： `Device# debug netconf error`	NETCONF セッションのデバッグをイネーブルにします。
ステップ 4	**clear netconf** {**counters** | **sessions**} 例： `Device# clear netconf sessions`	NETCONF 統計カウンタおよび NETCONF セッションをクリアし、関連するリソースを解放し、ロックを解除します。

NETCONF の設定例

例：NETCONF ネットワーク マネージャ アプリケーションの設定

次に、NETCONF を SSH サブシステムとして呼び出すように、NETCONF ネットワーク マネージャ アプリケーションを設定する例を示します。

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

NETCONF セッションの確立後すぐに、<hello> を含む次のような XML 文書を送信することによって、サーバの機能を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
    <hello>
      <capabilities>
        <capability>
            urn:ietf:params:xml:ns:netconf:base:1.0
          </capability>
          <capability>
            urn:ietf:params:ns:netconf:capability:startup:1.0
          </capability>
       </capabilities>
    <session-id>4<session-id>
</hello>]]>]]>
```

クライアントは、<hello> を含む XML 文書を送信して応答します。

```
<?xml version="1.0" encoding="UTF-8"?>
 <hello>
   <capabilities>
       <capability>
           urn:ietf:params:xml:ns:netconf:base:1.0
     </capability>
    </capabilities>
</hello>]]>]]>
```

次の XML 文字列を使用して、NETCONF ネットワーク マネージャ アプリケーションが NETCONF 通知を送受信できるようにします。

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>
```

NETCONF ネットワーク マネージャ アプリケーションの NETCONF 通知の送信または受信を停止するには、次の XML 文字列を使用します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>
```

例：NETCONF セッションのモニタリング

次に、**show netconf counters** コマンドからの出力例を示します。

```
Device# show netconf counters
NETCONF Counters
Connection Attempts:0: rejected:0 no-hello:0 success:0
Transactions
        total:0, success:0, errors:0
detailed errors:
        in-use 0        invalid-value 0         too-big 0
        missing-attribute 0     bad-attribute 0         unknown-attribute 0
        missing-element 0       bad-element 0   unknown-element 0
        unknown-namespace 0     access-denied 0         lock-denied 0
        resource-denied 0       rollback-failed 0       data-exists 0
        data-missing 0  operation-not-supported 0       operation-failed 0
        partial-operation 0
```

次に、**show netconf session** コマンドからの出力例を示します。

```
Device# show netconf session
(Current | max) sessions:   3 | 4
Operations received: 100                Operation errors: 99
Connection Requests: 5                  Authentication errors: 2   Connection Failures: 0
ACL dropped : 30
Notifications  Sent: 20
```

show netconf schema コマンドの出力は、NETCONF 要求およびその要求に対する応答のエレメント構造を表します。 このスキーマは、適切な NETCONF 要求の作成およびその要求に対する応答の解析に使用できます。 スキーマのノードについては RFC 4741 で規定されています。 次に、**show netconf schema** コマンドからの出力例を示します。

```
Device# show netconf schema
New Name Space 'urn:ietf:params:xml:ns:netconf:base:1.0'
<VirtualRootTag> [0, 1] required
  <rpc-reply> [0, 1] required
    <ok> [0, 1] required
    <data> [0, 1] required
    <rpc-error> [0, 1] required
      <error-type> [0, 1] required
      <error-tag> [0, 1] required
      <error-severity> [0, 1] required
      <error-app-tag> [0, 1] required
      <error-path> [0, 1] required
      <error-message> [0, 1] required
      <error-info> [0, 1] required
        <bad-attribute> [0, 1] required
        <bad-element> [0, 1] required
        <ok-element> [0, 1] required
        <err-element> [0, 1] required
```

```
      <noop-element> [0, 1] required
      <bad-namespace> [0, 1] required
      <session-id> [0, 1] required
<hello> [0, 1] required
  <capabilities> 1 required
    <capability> 1+ required
<rpc> [0, 1] required
  <close-session> [0, 1] required
  <commit> [0, 1] required
    <confirmed> [0, 1] required
    <confirm-timeout> [0, 1] required
  <copy-config> [0, 1] required
    <source> 1 required
      <config> [0, 1] required
        <cli-config-data> [0, 1] required
          <cmd> 1+ required
        <cli-config-data-block> [0, 1] required
        <xml-config-data> [0, 1] required
          <Device-Configuration> [0, 1] required
            <> any subtree is allowed
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <delete-config> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
  <discard-changes> [0, 1] required
  <edit-config> [0, 1] required
    <target> 1 required
      <candidate> [0, 1] required
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
    <default-operation> [0, 1] required
    <test-option> [0, 1] required
    <error-option> [0, 1] required
    <config> 1 required
      <cli-config-data> [0, 1] required
        <cmd> 1+ required
      <cli-config-data-block> [0, 1] required
      <xml-config-data> [0, 1] required
        <Device-Configuration> [0, 1] required
          <> any subtree is allowed
  <get> [0, 1] required
    <filter> [0, 1] required
      <config-format-text-cmd> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-text-block> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-xml> [0, 1] required
      <oper-data-format-text-block> [0, 1] required
        <show> 1+ required
      <oper-data-format-xml> [0, 1] required
        <show> 1+ required
  <get-config> [0, 1] required
    <source> 1 required
      <config> [0, 1] required
        <cli-config-data> [0, 1] required
          <cmd> 1+ required
        <cli-config-data-block> [0, 1] required
        <xml-config-data> [0, 1] required
          <Device-Configuration> [0, 1] required
            <> any subtree is allowed
      <candidate> [0, 1] required
```

```
      <running> [0, 1] required
      <startup> [0, 1] required
      <url> [0, 1] required
    <filter> [0, 1] required
      <config-format-text-cmd> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-text-block> [0, 1] required
        <text-filter-spec> [0, 1] required
      <config-format-xml> [0, 1] required
<kill-session> [0, 1] required
  <session-id> [0, 1] required
<lock> [0, 1] required
  <target> 1 required
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
<unlock> [0, 1] required
  <target> 1 required
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
<validate> [0, 1] required
  <source> 1 required
    <config> [0, 1] required
      <cli-config-data> [0, 1] required
        <cmd> 1+ required
      <cli-config-data-block> [0, 1] required
      <xml-config-data> [0, 1] required
        <Device-Configuration> [0, 1] required
          <> any subtree is allowed
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
<notification-on> [0, 1] required
<notification-off> [0, 1] required
```

NETCONF 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
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
NETCONF コマンド：コマンド構文、コマンドモード、コマンド履歴、デフォルト、使用に関する注意事項、および例	『*Cisco IOS Cisco Networking Services Command Reference*』
セキュリティおよび IP アクセス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上の注意事項、および例	『*Cisco IOS Security Command Reference*』

標準および RFC

標準/RFC	タイトル
RFC 4251	『*The Secure Shell (SSH) Protocol Architecture*』
RFC 4252	『*The Secure Shell (SSH) Authentication Protocol*』
RFC 4741	『*NETCONF Configuration Protocol*』
RFC 4744	『*Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)*』

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

NETCONF 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 *1*：***NETCONF*** の機能情報

機能名	リリース	機能情報
NETCONF	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRA 12.2(33)SXI 12.4(9)T	NETCONF プロトコルは、ネットワーク デバイスの管理、設定データの取得、および新しい設定データのアップロードと操作の簡単なメカニズムを定義します。 NETCONF では、設定データおよびプロトコル メッセージとして拡張可能マークアップ言語（XML）ベースのデータ符号化を使用します。 この機能により、**clear netconf**、**debug netconf**、**show netconf** の各コマンドが導入または変更されました。
NETCONF XML PI	Cisco IOS XE Release 3.8S 15.3(1)S 15.3(1)T	NETCONF プロトコルが拡張され、**clear netconf**、**debug netconf**、**show netconf** コマンドを含むすべての Cisco IOS EXEC コマンドの形式属性のサポートが追加されました。

用語集

BEEP：ブロック拡張可能交換プロトコル。 コネクション型非同期相互作用のための汎用アプリケーション プロトコル フレームワーク。

NETCONF：ネットワーク設定プロトコル。 ネットワーク デバイスの管理、設定データの取得、および新しい設定データのアップロードと操作の簡単なメカニズムを定義するプロトコル。

SASL：Simple Authentication and Security Layer。 接続ベースのプロトコルに認証サポートを追加するためのインターネット標準方式。 SASL を、セキュリティ アプライアンスと Lightweight Directory Access Protocol（LDAP）サーバとの間で使用して、ユーザ認証を強化できます。

SSHv2：セキュア シェル バージョン 2。 SSH は、信頼性の高いトランスポート層の上部で実行され、強力な認証機能と暗号化機能を提供します。 SSHv2 を使用すると、別のコンピュータにネットワークを介して安全にアクセスして安全にコマンドを実行できるようになります。

TLS：トランスポート層セキュリティ。 相互認証、完全性のためのハッシュの使用、プライバシー保護のための暗号化を可能にすることで、クライアントとサーバとの間にセキュアな通信を実現

するアプリケーション レベルのプロトコルです。 TLS では、証明書、公開キー、および秘密キーを使用します。

XML：拡張可能マークアップ言語。 World Wide Web Consortium（W3C）によって管理されている、情報構造を指定するマークアップ言語を作成するための構文を定義する標準。 情報構造は、情報の外観（太字、イタリック体など）ではなく、情報のタイプ（加入者名やアドレスなど）を定義します。 外部のプロセスでこれらの情報構造を操作し、さまざまなフォーマットで公開することができます。 XML では、独自にカスタマイズしたマークアップ言語を定義できます。