



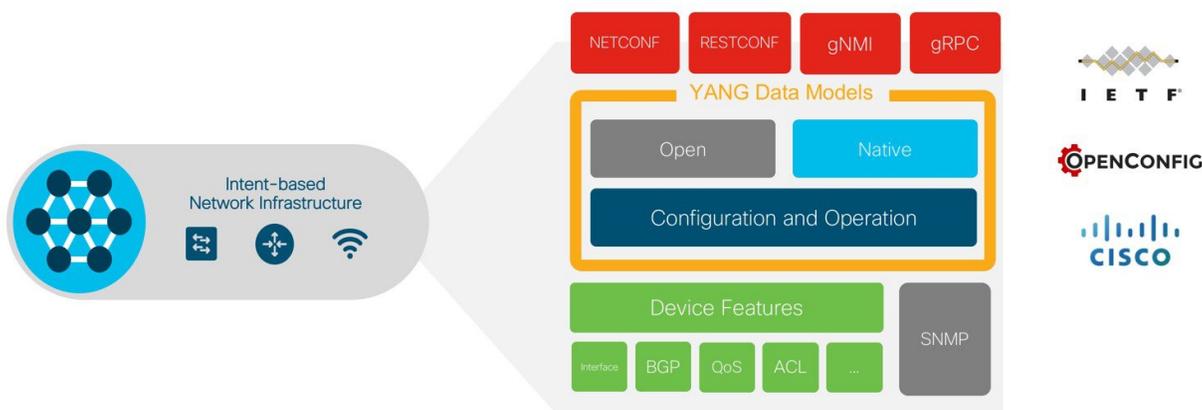
## Catalyst 9800 プログラマビリティとテレメトリ導入ガイド 16.10

機能の概要	2
コンフィギュレーションの概要	2
NETCONF	3
RESTCONF (ベータ版、サポート対象外)	5
gRPC と gNMI (ベータ版、サポート対象外)	7
YANG データ モデル	8
YangExplorer ツール	10
Ansible Automation	16
テレメトリ	21
まとめ	25
関連リソース	25
参照先	26
ご質問がある場合	27

改訂：2019年2月2日

## 機能の概要

Catalyst 9800 IOS-XE ベースのワイヤレス LAN コントローラにはプログラミングによる設定用のいくつかのオプションがあります。WLCを設定するための従来の方法としてはCLIまたはWebUIがありますが、拡張されて、プログラミング対応インターフェイスが含まれるようになりました。これらのプログラミング対応インターフェイスにはNETCONF、RESTCONF、gNMI/gRPCなどのプロトコルが含まれます。YANG データモデルは、プログラミング対応インターフェイスを介してアクセスできるデータを定義します。また、これらのモデルにはいくつかの種類があります。Cisco IOS XE の機能はネイティブ データ モデル内で定義され、標準的な機能やベンダー非依存機能はオープン データ モデル内で定義されます。どちらのモデルも多くのタスクに使用できますが、IOS-XE に固有の機能はネイティブモデルでのみ使用できます。



## コンフィギュレーションの概要

使用しているインターフェイスに関係なく、Catalyst 9800 では次の設定が必要です。

```
configure terminal
aaa new-model
aaa authentication login default local
aaa authorization exec default local
aaa session-id common
exit
write memory
```

さらに、ユーザアカウントも必要です。ユーザ名「netconf」、パスワード「netconf」を使用し、次のコマンドで新しいユーザアカウントを作成します。

```
username netconf privilege 15 password 0 netconf
username restconf privilege 15 password 0 restconf
```

専用の NETCONF または RESTCONF のユーザアカウントが必要でない場合、代わりに既存または事前設定済の admin ユーザとパスワードを使用することができます。

## 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」ために IETF によって定義されたプロトコルです。NETCONF の動作は XML エンコーディングを使用してリモートプロシージャ コール (RPC) レイヤの最上部で認識され、ネットワーク デバイス上の設定を編集および照会するための基本的な一連の動作を提供します。

### NETCONF の設定

NETCONF インターフェイスを有効にするには、次のコマンドを入力します。

```
configure terminal
netconf-yang
exit
```

### NETCONF YANG モデル

デフォルト ポート 830 上のデバイスに接続することで NETCONF 機能交換を取得できます。機能交換には、デバイスがサポートしていて使用可能なすべての YANG データ モデルがリストされます。以降の項で詳しく説明する YangExplore や pYANG のようなツールを使用して、これらの YANG モジュールをデバイスからダウンロードし、さらに分析することができます。

```
user@canyon-server:~$ ssh netconf@10.10.10.223 -p 830
netconf@10.10.10.223's password:
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
<capability>urn:ietf:params:netconf:capability:notification:1.0</capability>
<capability>urn:ietf:params:netconf:capability:interleave:1.0</capability>
```

### NETCONF ステータスの確認

NETCONF インターフェイスが動作可能であることを確認するには、**show platform software yang-management process** コマンドを実行します。NETCONF SSHD 「ncsshd」プロセスが実行されていることを確認します。

```

jcohoe-vewlc-2#show platform software yang-management process
confd                : Running
nesd                 : Running
syncfd              : Running
ncsshd ←            : Running
dmiautha            : Running
nginx                : Running
ndbmand             : Running
pubd                 : Running
gnmib                : Not Running

jcohoe-vewlc-2#

```

## NETCONF XML RPC ペイロードの例

次に示す XML RPC ペイロードは YangExplorer ツール内または NETCONF インターフェイスを介して XML ペイロードを送信できる Python スクリプトなどの他のツール内から生成、変更、送信できます。次に、XML ペイロードの WLC による作成、確認、および削除の例を示します。これらを使用すると、NETCONF インターフェイスを介して WLAN の作成、確認、および削除をすばやく簡単に行えます。

### WLAN の作成

次に示す XML RPC を NETCONF インターフェイスを介して送信すると、定義されたパラメータを使用して WLAN を作成できます。この場合、WLAN ID は 4、SSID は「open」です。

```

<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="6">
<edit-config xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <target>
    <running/>
  </target>
  <test-option>test-then-set</test-option>
  <error-option>rollback-on-error</error-option>
  <config>
    <wlan-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-wlan-cfg">
      <wlan-cfg-entries>
        <wlan-cfg-entry>
          <profile-name>open</profile-name>
          <vap-id>4</vap-id>
          <security-wifi-sec>>false</security-wifi-sec>
          <rsn-ie-enabled>>false</rsn-ie-enabled>
          <rsn-cipher-suite-aes>>false</rsn-cipher-suite-aes>
          <auth-key-mgmt-suite8021x>>false</auth-key-mgmt-suite8021x>
          <client-session-timeout>1800</client-session-timeout>
          <id-data>
            <profile-name>open</profile-name>
            <ssid>open</ssid>
            <status>>true</status>
          </id-data>
          <is-remote-lan>>false</is-remote-lan>
        </wlan-cfg-entry>
      </wlan-cfg-entries>
    </wlan-cfg-data>
  </config></edit-config></rpc>

```

## WLAN の確認

次に示す XML RPC を NETCONF インターフェイスを介して送信すると、ID 4 を持つ WLAN を確認できます。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <wlan-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-wlan-cfg">
        <wlan-cfg-entries>
          <wlan-cfg-entry>
            <vap-id>4</vap-id>
          </wlan-cfg-entry>
        </wlan-cfg-entries>
      </wlan-cfg-data>
    </filter>
  </get>
</rpc>
```

## WLAN の削除

次に示す XML RPC を NETCONF インターフェイスを介して送信すると、ID 4 を持つ WLAN を削除できます。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <wlan-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-wlan-cfg">
        <wlan-cfg-entries xc:operation="delete">
          <wlan-cfg-entry>
            <vap-id>4</vap-id>
          </wlan-cfg-entry>
        </wlan-cfg-entries>
      </wlan-cfg-data>
    </config>
  </edit-config>
</rpc>
```

## RESTCONF (ベータ版、サポート対象外)

RESTCONF は構造化 XML または JSON と YANG データ モデルを使用して REST ライクな API を提供します。これによりネットワークデバイスへのプログラミングでのアクセスが可能になります。RESTCONF API は HTTPS メソッドを使用します。Catalyst 9800 IOS XE の実装は、GET、PATCH、PUT、POST、DELETE、HEAD の RESTCONF 動作をサポートします。

## RESTCONF の設定

RESTCONF インターフェイスを有効にするには、次のコマンドを入力します。

```
configure terminal
ip http secure-server
restconf
exit
```

## RESTCONF YANG モデル

サポート対象の YANG データ モデルのリストは、URI `restconf/data?fields=ietf-yang-library:modules-state/module` に GET 要求を送信することで RESTCONF インターフェイスから取得できます。

次に例を示します。

```
curl -k -u username:password https://10.10.10.223/restconf/data?fields=ietf-yang-library:modules-state/module
```

```
user@canyon-server:~$ curl -k -u jcoho:jcoho https://10.10.10.223/restconf/data?fields=ietf-yang-library:modules-state/module
<data xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
  <modules-state xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
    <module>
      <name>ATM-FORUM-TC-MIB</name>
      <revision></revision>
      <schema>https://10.10.10.223:443/restconf/tailf/modules/ATM-FORUM-TC-MIB</schema>
      <namespace>urn:ietf:params:xml:ns:yang:smiv2:ATM-FORUM-TC-MIB</namespace>
      <conformance-type>import</conformance-type>
    </module>
```

```
<module>
  <name>Cisco-IOS-XE-wireless-access-point-oper</name>
  <revision>2018-06-28</revision>
  <schema>https://10.10.10.223:443/restconf/tailf/modules/Cisco-IOS-XE-wireless-access-point-oper/2018-06-28</schema>
  <namespace>http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper</namespace>
  <conformance-type>implement</conformance-type>
</module>
<module>
  <name>Cisco-IOS-XE-wireless-access-point-oper</name>
  <revision>2018-07-09</revision>
  <schema>https://10.10.10.223:443/restconf/tailf/modules/Cisco-IOS-XE-wireless-access-point-oper/2018-07-09</schema>
  <namespace>http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper</namespace>
  <conformance-type>implement</conformance-type>
</module>
```

YANG データ モデルを含んだ URI に別の GET 要求を送信すると特定の YANG モジュールをダウンロードできます。たとえば、Cisco-IOS-XE-wireless-access-point-oper モデルをダウンロードするには、次に示すスクリーンショットのように `restconf/tailf/modules/Cisco-IOS-XE-wireless-access-point-oper/2018-06-28` へ GET 要求を送信します。

```
curl -s -k -u jcoho:jcoho https://10.10.10.223/restconf/tailf/modules/Cisco-IOS-XE-wireless-access-point-oper/2018-06-28 -o Cisco-IOS-XE-wireless-access-point-oper.yang
user@canyon-server:~$ pyang -f tree Cisco-IOS-XE-wireless-access-point-oper.yang --tree-depth 2
Cisco-IOS-XE-wireless-access-point-oper.yang:6: error: module "Cisco-IOS-XE-wireless-ap-types" not found in search path
Cisco-IOS-XE-wireless-access-point-oper.yang:9: error: module "Cisco-IOS-XE-wireless-enum-types" not found in search path
Cisco-IOS-XE-wireless-access-point-oper.yang:12: error: module "Cisco-IOS-XE-wireless-types" not found in search path
Cisco-IOS-XE-wireless-access-point-oper.yang:6032 (at Cisco-IOS-XE-wireless-access-point-oper.yang:2641): error: the key "wtp-mac" does not reference an existing leaf
Cisco-IOS-XE-wireless-access-point-oper.yang:6016: error: the key "wtp-mac" does not reference an existing leaf
Cisco-IOS-XE-wireless-access-point-oper.yang:6048: error: the key "radio-mac" does not reference an existing leaf
Cisco-IOS-XE-wireless-access-point-oper.yang:6055: error: the key "radio-mac" does not reference an existing leaf
Cisco-IOS-XE-wireless-access-point-oper.yang:6062: error: the key "radio-mac" does not reference an existing leaf
module: Cisco-IOS-XE-wireless-access-point-oper
+--ro access-point-oper-data
  +--ro ap-history* [ethernet-mac]
  |   ...
  +--ro wireless-interface-data!
  |   ...
  +--ro radio-oper-data* [wtp-mac radio-slot-id]
  |   ...
  +--ro qos-client-data* [client-mac]
  |   ...
  +--ro capwap-data* [wtp-mac]
  |   ...
  +--ro ap-name-mac-map* [wtp-name]
```

上のステップでは、YANG データ モデル内にいくつかのデータが含まれている詳細情報に対して「pyang」が使用されています。この例では、`ap-history`、`wireless-interface-data`などが「`access-point-oper-data`」コンテナ内のセクションです。

## RESTCONF ステータスの確認

RESTCONF インターフェイスが動作可能であることを確認するには、`show platform software yang-management process` コマンドを実行します。「`nginx`」プロセスが実行されていることを確認します。

```
jcohoe-vewlc-2#show platform software yang-management process
confd          : Running
nesd           : Running
syncfd         : Running
ncsshd         : Running
dmiauthd       : Running
nginx          : Running
ndbmana        : Running
pubd           : Running
gnmib          : Not Running

jcohoe-vewlc-2#
```

## gRPC と gNMI (ベータ版、サポート対象外)

gNMI は Google によって開発された gRPC ネットワーク管理インターフェイスです。gNMI はネットワークデバイスの設定をインストール、操作、および削除し、また、運用データの表示も実行するメカニズムです。gNMI を通じて提供されるコンテンツは YANG を使用してモデル化できます。gRPC はクラウドサーバと通信するモバイルクライアントを使用して低遅延で拡張可能な配布を実現するために Google によって開発されたリモートプロシージャコールです。gRPC は gNMI を伝送し、データと動作要求を公式化して送信する手段を提供します。

### gNMI の設定

gNMI インターフェイスを有効にするには、次のコマンドを入力します。

```
configure terminal
gnmi-yang
gnmi-yang server
gnmi-yang port 50052
exit
write memeory
```

### gNMI の動作

gNMI インターフェイスを介して次の動作がサポートされています。

gNMI GetRequest

gNMI SetRequest

このインターフェイスの利用に関するその他の詳細は、「その他のリソース」の項でリンクされている『Programmability Configuration Guide』から入手できます。

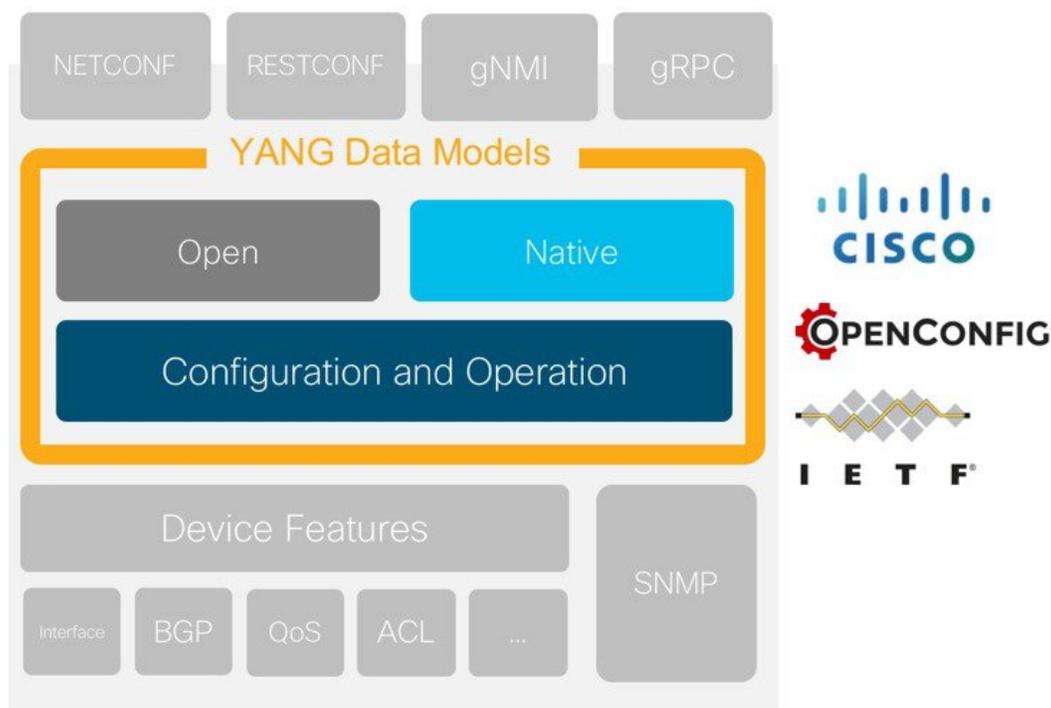
### gNMI ステータスの確認

gNMI インターフェイスが動作可能であることを確認するには、`show gnmi-yang state` コマンドを実行します。状態が `enabled` で、ステータスが `up` になっていることを確認します。

```
WLC#show gnmi-yang state
State           Status
-----
Enabled         Up
```

## YANG データ モデル

YANG は NETCONF 用のデータ モデリング言語です。IOS XE デバイス内の YANG モデルは、送受信するデータの構築方法を説明するように定義されています。YANG 標準は RFC6020 で定義されました。使用されている YANG モデルにはネイティブとオープン の 2 つの主要タイプがあります。モデルはさらに設定モデルと動作モデルのいずれかに分類されます。設定モデルはプログラミングによる設定に使用でき、動作モデルはテレメトリで使用できます。



## ネイティブ モデル

ワイヤレス用のネイティブ モデルは設定と運用の 2 つの主要カテゴリに分類できます。設定モジュールには関連機能の設定情報が含まれ、運用モデルは機能に関する実行時間と運用データを提供します。

# Cisco-IOS-XE-wireless: Modules

<b>Config:</b>	mesh
ap	mobility
apf	mstream
cts-sxp	rf
dot11	rfid
fabric	rogue
flex	rrm
fqdn	security
general	site
location	wlan

<b>Oper:</b>	mobility
access-point	nmsp
client	rf-profile
fqdn	rfid
lisp-agent	rogue
mcast	rrm
mesh	

## ネイティブ設定モデル

Config Module	Data Model Details
Ap	AP configuration: tags, policy tags, rf tags, site tags
Apf	Global dot11 parameters: country, association request limit, blacklisting
Cts-sxp	CTS SXP configuration: connection information, SXP mode, password, profile name, hold times, retry periods
Dot11	802.11 configuration: TX Power, 802.11n/802.11ac features, admin state, data rates
Fabric	Fabric configuration: VNID name, SDA config, fabric profile
Flex	Flex configuration: RLAN configuration, VLAN ACL's
FQDN	FQDN configurations: url-filter, domain names, filter action
general	Misc WLC configurations: AP join configs, wireless management interfaces, multicast, WSA config
location	Location configurations: NMSP, CMX cloud, server URL, timeouts, measurement intervals
mesh	Wireless Mesh configurations: mesh profile, psk, aaa method, threshold values
mobility	Mobility configurations: mode, config, peer config, group config, guest mode
mstream	Multicast configurations: group, url, group name, stream groups, priority
rf	RF configurations: allowed channels, rf-profile, data rates config, DCA
rfid	RFID configurations: timeouts, RSSI expiry, notify thresholds
rogue	Rogue configurations: rogue global parameters, SSID rules, clients, ignore list
rrm	RRM configuration: intervals, grouping algorithm, static members
security	AP Security configurations: AP LSC config, CA trustpoint, key size
site	Site configurations: site tag, rlan ports, master AP, join profile, BLE, AP PCAP, AP config profiles
wlan	WLAN configuration: aaa, dhcp, umbrella, 802.11v, flex profiles, WLAN SSID, 802.1x security

## ネイティブ動作モデル

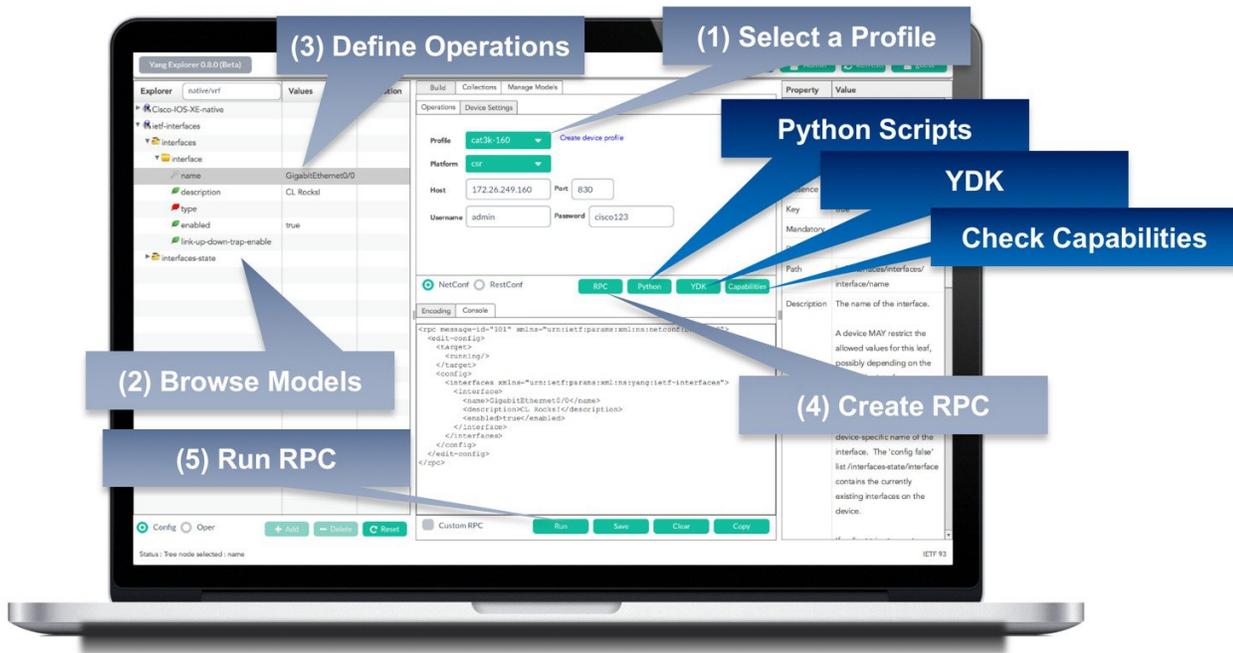
Operational Modules	Data Model Details
access-point	Join failures, discovery, certificates, country, QoS, ATF, radio power
client	Session details, eapol, wlan id, delete reasons, association details
FQDN	DNS snooping, learned IP's, timestamps, actions applied
LISP	Fabric capabilities, lisp agent, tcp and udp failure counts, map registers
mcast	Multicast
mesh	Mesh
mobility	Mobility
nmsp	NMSP Location
rf-profile	RF Profiles
rfid	RFID
rogue	Rogues
rrm	Radio Resource Management

## オープンモデル

シスコのネイティブ設定と動作モデルの他に、デバイス上でサポートされている他の YANG モデルがいくつかあります。NETCONF および RESTCONF の YANG ライブラリを介した機能交換では、これらのモデルのすべてがリストされます。SNMP MIB、IETF、および OpenConfig 用のモデルがあります。これらのモデルはネイティブモデルと同じ方法で使用できますが、デバイスで利用可能な機能のうちの限定された機能またはそれらのサブセットが提供されます。そのため、可能な限りシスコ ネイティブ モデルを利用することをお勧めします。

## YangExplorer ツール

YangExplorer は NETCONF インターフェイスおよび RESTCONF インターフェイスとのやり取りに使用されるツールです。GitHub で利用でき、XML RPC ペイロードを生成してデバイスに送信するのに使用できます。



<https://github.com/CiscoDevNet/yang-explorer>

## YangExplorer のインストール

<https://github.com/CiscoDevNet/yang-explorer> にある GitHub サイトでの指示に従ってインストールを完了します。Mac と Linux 用の詳細手順が入手できます。

## ###2. Installation ###2.1 First time installation #####Prerequisite:

- MAC, Linux (not supported on Windows)
- python 2.7
- pip package manager (<https://pip.pypa.io/en/stable/installing/>)

If already installed, make sure that pip / setuptools are upto date (commands may vary)

```
pip install --upgrade pip
```

```
Ubuntu: sudo pip install --upgrade setuptools
```

- virtualenv (recommended)

```
Ubuntu: sudo apt-get install python-virtualenv
```

```
Fedora: sudo dnf install python-virtualenv
```

```
MAC: sudo pip install virtualenv
```

- graphviz (<http://www.graphviz.org/Download.php>)

```
Ubuntu: sudo apt-get install graphviz
```

```
Fedora: sudo dnf install graphviz
```

```
MAC: brew install graphviz
```

- Browser with latest flash plugin (tested with google chrome)

## #####Download and install:

```
git clone https://github.com/CiscoDevNet/yang-explorer.git
cd yang-explorer
bash setup.sh
```

Note: sudo may be required if you do not use virtualenv.

## YangExplorer Day 0 設定

YangExplorer がインストールされ、実行された後は、Catalyst 9800 とやり取りする前にいくつかのタスクを完了する必要があります。[Create device profile] を選択し、ユーザ名とパスワードに「admin」を使用して [Django administration] ページに移動します。[Add Device Profile] ページがロードされます。このページでは、デバイスに関する情報を追加する必要があります。プロファイル名を入力し、デバイス タイプを選択して、NETCONF IP、RESTCONF IP、ユーザ名およびパスワードを入力します。

# YangExplorer – Initial Configuration

- Select “Create Device Profile” from main Yang Explorer page
- Use “admin” for both username and password for the Django administration

Enter details for the cat9800 device including:  
Profile name, device type, user type, NETCONF IP,  
username, password, and description

デバイスが追加された後、メインの YangExplorer ウィンドウを更新すると [Profile] ドロップダウンを選択したときに使用可能なデバイスが表示されます。

セットアップが正常であることを確認するには、[Capabilities] ボタンを選択します。選択したデバイスに YangExplorer が接続され、そのデバイスでサポートされている YANG モジュールのリストが返されます。

Profile **cat9800** [Create device profile](#)

Platform **other**

Host  Port

Username  Password

NetConf  RestConf

[RPC](#) [Python](#) [YDK](#) [Capabilities](#)

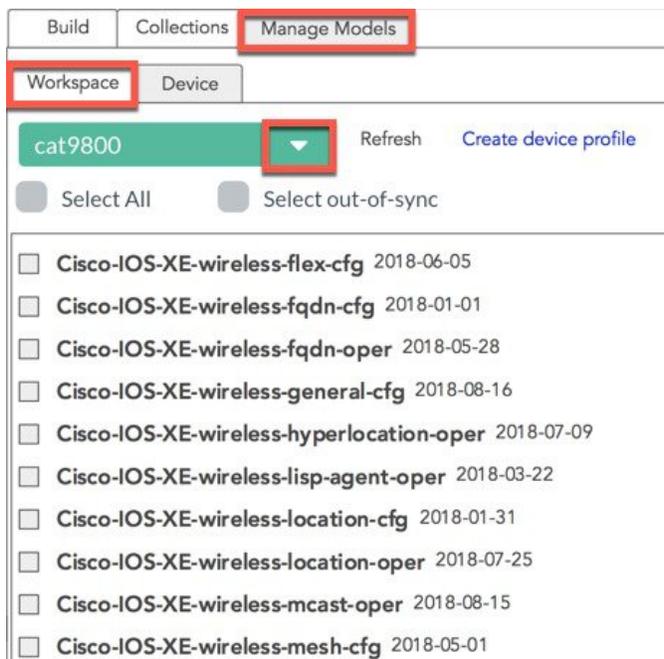
Encoding

```
http://cisco.com/ns/yang/Cisco-IOS-XE-wccp?module=Cisco-IOS-XE-wccp&revision=2017-11-27
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper?module=Cisco-IOS-XE-wireless-access-point-oper&revision=2018-06-28
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-ap-cfg?module=Cisco-IOS-XE-wireless-ap-cfg&revision=2018-05-10
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-ap-types?module=Cisco-IOS-XE-wireless-ap-types&revision=2018-06-15
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-apf-cfg?module=Cisco-IOS-XE-wireless-apf-cfg&revision=2018-08-13
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-client-oper?module=Cisco-IOS-XE-wireless-client-oper&revision=2018-09-06
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-client-types?module=Cisco-IOS-XE-wireless-client-types&revision=2018-08-08
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-cts-sxp-cfg?module=Cisco-IOS-XE-wireless-cts-sxp-cfg&revision=2018-04-09
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-cts-sxp-oper?module=Cisco-IOS-XE-wireless-cts-sxp-oper&revision=2018-08-07
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-dot11-cfg?module=Cisco-IOS-XE-wireless-dot11-cfg&revision=2018-06-13
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-enum-types?module=Cisco-IOS-XE-wireless-enum-types&revision=2018-05-18
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-events-oper?module=Cisco-IOS-XE-wireless-events-oper&revision=2018-06-11
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-fabric-cfg?module=Cisco-IOS-XE-wireless-fabric-cfg&revision=2018-03-08
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-flex-cfg?module=Cisco-IOS-XE-wireless-flex-cfg&revision=2018-06-05
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-fqdn-cfg?module=Cisco-IOS-XE-wireless-fqdn-cfg&revision=2018-01-01
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-fqdn-oper?module=Cisco-IOS-XE-wireless-fqdn-oper&revision=2018-05-28
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-general-cfg?module=Cisco-IOS-XE-wireless-general-cfg&revision=2018-08-16
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-hyperlocation-oper?module=Cisco-IOS-XE-wireless-hyperlocation-oper&revision=2018-07-09
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-lisp-agent-oper?module=Cisco-IOS-XE-wireless-lisp-agent-oper&revision=2018-03-22
http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-location-cfg?module=Cisco-IOS-XE-wireless-location-cfg&revision=2018-01-31
```

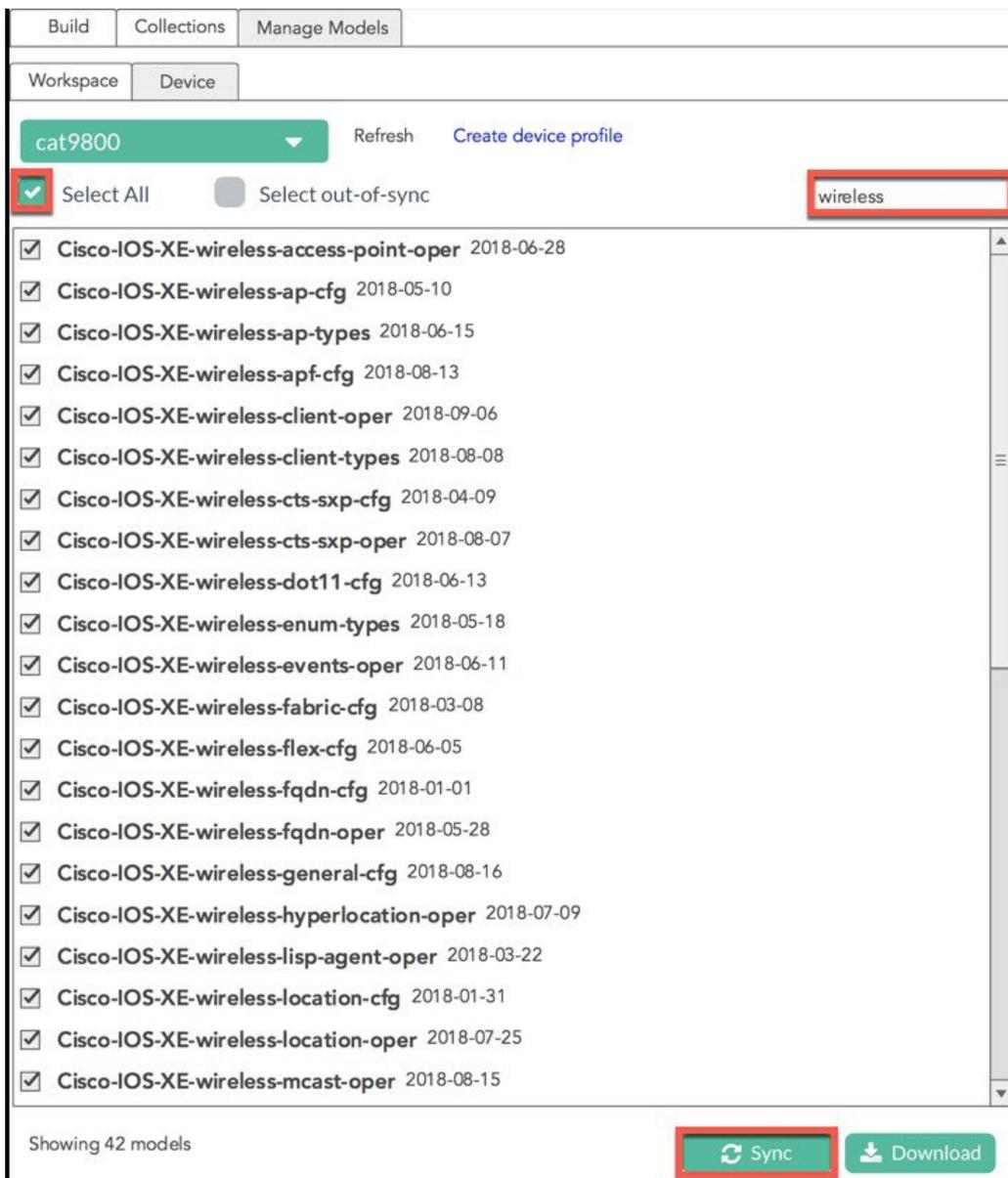
## YangExplorer からの YANG モデルのダウンロード

YangExplorer 内で、デバイスから YANG モジュールをダウンロードできます。同期してモデルにサブスクライブする手順に従います。

1. [Manage Models] タブを選択します。
2. [Device] タブを選択します。
3. 下矢印を選択してデバイス プロファイルを選択します。



4. ワイヤレス モジュールのみを表示するには、検索ボックスに「wireless」と入力します。
5. [Select All] チェックボックスをオンにしてすべてのワイヤレス モジュールを選択します。
6. [Sync] を選択してデバイスからモジュールをダウンロードします。



について

- ダウンロードの完了後に [Subscribe] を選択すると、YangExplorer の [Explorer] タブでモデルが使用可能になります。



## Ansible Automation

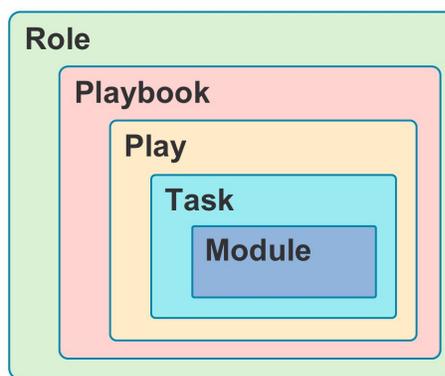
Ansible はソフトウェアのプロビジョニング、設定、および管理を自動化する一般的で使いやすいオープンソース ソフトウェアスイートです。SSH、NETCONF、およびその他さまざまなプロトコルを介してデバイスに接続してそれらを制御します。Ansible はエージェントレスです。つまり、アクセス可能な API やインターフェイスが必要である以外は、ターゲットデバイスでのインストールや要件がありません。事実上、最小であり、リモートデバイスとやり取りする

ためのセキュアで信頼性の高い手段を提供します。適応性が高いため、一般に複雑なワークフローを実行するために他の自動化ツールとともに使用されます。次に、基本的な Day 0 設定タスクの実行に Ansible を使用する例を示します。

Ansible には、総合的なソリューションを実現するために連携するコンポーネントがいくつかあります。再利用可能なモジュールとコール可能なスタンドアロン スクリプトが存在します。タスクでモジュールをコールし、アクションを実行します。複数のタスクがある場合は、プレイを使用してタスクを順番にコールできます。プレイが複数ある場合は、プレイブックが使用されます。最後に、一連のプレイブックがロールです。

## Ansible Taxonomy

- Role: a set of Playbooks ( )
- Playbook: repeatable standard config
- Play: a set of tasks
- Task: single action that references a module
- Module: reusable, standalone scripts



## Ansible の例 : NETCONF-YANG と RESTCONF の有効化

ホスト ファイルには接続の詳細と、クレデンシャルなどのデバイス固有の情報が含まれています。この例では、SSH ログインを許可するように Catalyst 9800 がすでに設定されていて、イネーブルパスワードも設定されていることが想定されます。ホスト ファイルと YAML ファイルを一緒に使用してタスクが実行されます。この例では、SSH を介して CLI に接続してイネーブルモードに移行し、必要な IOS コマンドを実行して netconf-yang を有効にし、AAA 要求をセットアップします。

## Day 0 設定の Ansible ホスト ファイル

この hosts.txt ファイルの例には、デバイスへの接続を正常に確立するために必要な変数が含まれています。

```
[all:vars]
ansible_connection=network_cli
ansible_network_os=ios
ansible_user=admin
ansible_password=Cisco123
ansible_become=yes
ansible_become_method=enable
ansible_become_pass=Cisco123
[cat9800]
cat9800-1 ansible_host=10.10.10.224
```

```
cat hosts.txt
[all:vars]
ansible_connection=network_cli
ansible_network_os=ios
ansible_user=admin
ansible_password=Cisco123
ansible_become=yes
ansible_become_method=enable
ansible_become_pass=Cisco123

[cat9800]
cat9800-1 ansible_host=10.10.10.223
```

## NETCONF と RESTCONF を有効にする Ansible YAML 設定ファイル

この `enable_netconf_yang.yaml` YAML ファイルの例は、デバイス上で NETCONF インターフェイスを有効にし、ユーザの追加などの認証の前提条件を設定するために使用できます。

```
---
- hosts: cat9800
  gather_facts: no

  tasks:
    - ios_config:
      commands:
        - aaa new-model
        - aaa authorization exec default local
        - aaa authentication login default local
        - aaa session-id common
        - username netconf privilege 15 password 0 netconf
        - netconf-yang
        - ip http secure-server
        - restconf
      save_when: modified
```

```

cat enable_netconf_yang.yaml
---
- hosts: cat9800
  gather_facts: no

  tasks:
    - ios_config:
      commands:
        - aaa new-model
        - aaa authorization exec default local
        - aaa authentication login default local
        - aaa session-id common
        - username netconf privilege 15 password 0 netconf
        - netconf-yang
        - ip http secure-server
        - restconf
      save_when: modified

```

## NETCONF と RESTCONF 設定を確認する Ansible YAML 表示ファイル

この cat9800\_verify.yaml YAML ファイルの例では、2つの ios show コマンドを実行して netconf-yang と restconf が有効であることを確認し、実行されると出力が登録されて画面に表示されます。

```

- hosts: cat9800
  gather_facts: no

  tasks:
    - ios_command:
      commands:
        - show run | i netconf-yang
        - show run | i restconf
      register: show
    - debug: var=show.stdout_lines

```

```

cat cat9800_verify.yaml
---
- hosts: cat9800
  gather_facts: no

  tasks:
    - ios_command:
      commands:
        - show run | i netconf-yang
        - show run | i restconf
      register: show
    - debug: var=show.stdout_lines

```

## NETCONF と RESTCONF を有効にして確認するタスクの実行

「ansible-playbook」コマンドを使用すると、前述で定義したタスクを実行し、デバイス上でNETCONF-YANGインターフェイスを有効にすることができます。この例では、変数を定義してHost Key Checkingをfalseに設定し、SSHホストキーが検証されないようにします。実稼働環境ではアクセスされる側のデバイスの信頼性を確認することが重要ですが、このラボの例では使用しやすいようにチェックがfalseに設定されています。

コマンドラインから次のコマンドを実行して enable\_netconf\_yang.yaml 設定を実行します。

```
$ ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -i ./hosts.txt ./enable_netconf_yang.yaml
```

```
ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -i ./hosts.txt ./enable_netconf_yang.yaml
PLAY [cat9800] *****
TASK [ios_config] *****
changed: [cat9800-1]
PLAY RECAP *****
cat9800-1          : ok=1    changed=1    unreachable=0    failed=0
```

コマンドラインから次のコマンドを実行して cat9800\_verify.yaml 設定を実行します。

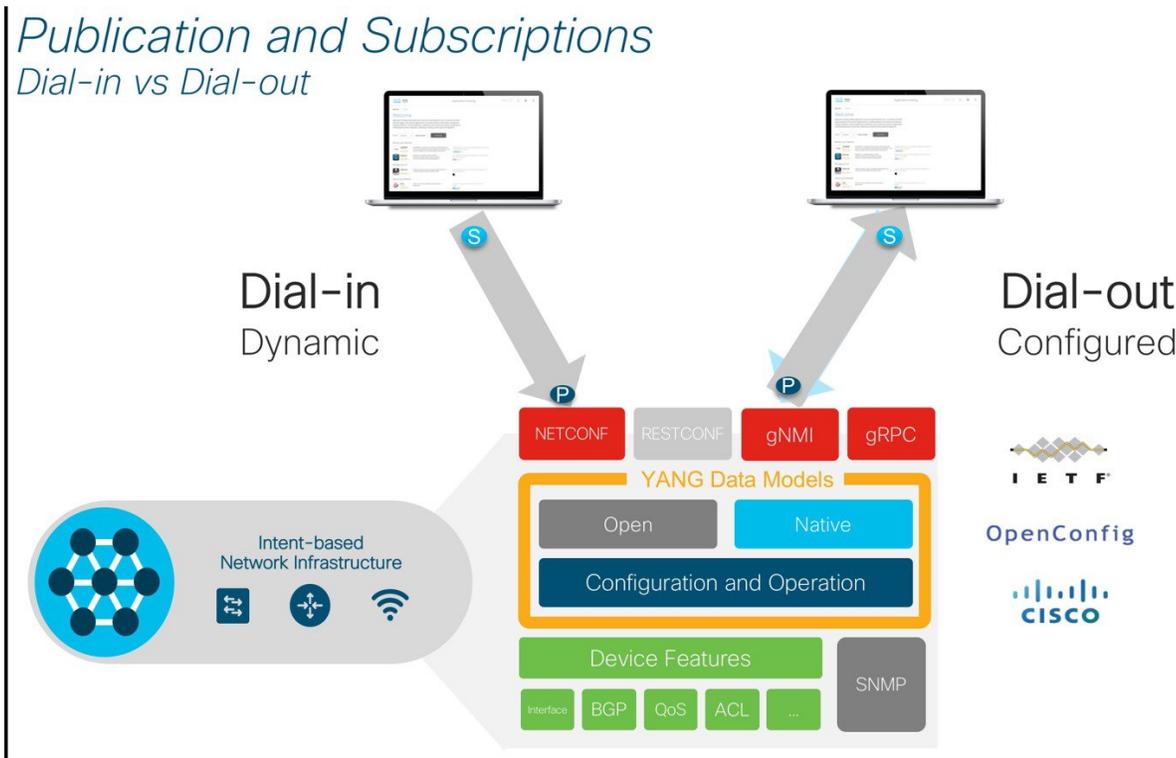
```
$ ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -i ./hosts.txt ./cat9800_verify.yaml
```

```
ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -i ./hosts.txt ./cat9800_verify.yaml
PLAY [cat9800] *****
TASK [ios_command] *****
ok: [cat9800-1]
TASK [debug] *****
ok: [cat9800-1] => {
  "show.stdout_lines": [
    [
      "netconf-yang"
    ],
    [
      "restconf"
    ]
  ]
}
PLAY RECAP *****
cat9800-1          : ok=2    changed=0    unreachable=0    failed=0
```

## テレメトリ

### ダイヤルイン ダイナミック サブスクリプションとダイヤルアウト設定済みサブスクリプション

NETCONF インターフェイスと gNMI インターフェイスの両方が、ダイヤルイン設定またはダイヤルアウト設定のいずれかでテレメトリ サブスクリプションをサポートします。IOS XE は NETCONF ダイヤルインをサポートし、gRPC はダイヤルアウト形式の設定をサポートします。ダイヤルインまたは「ダイナミック」サブスクリプションでは、サブスクリバがまずデバイスへのセッション接続を確立してから、データ モデルにサブスクライブする必要があります。テレメトリ データのストリーミングを維持するには、NETCONF セッションを確立したままにする必要があります。セッションが切断された場合は、テレメトリ サブスクリプションを手動で再確立する必要があります。ダイヤルアウトまたは「設定済み」サブスクリプションでは、ユーザによって設定がセットアップされるとデバイスがサブスクリプション設定を維持してサブスクライバにテレメトリを送信します。コレクタへのアクティブなセッションは必要ありません。



### NCC を使用したダイヤルイン NETCONF ダイナミック テレメトリのサブスクリプション

ダイヤルインテレメトリ サブスクリプションは <https://github.com/CiscoDevNet/ncc> から入手できるオープンソースの NCC ツールを使用して簡単に作成できます。レポジトリは git clone コマンドを使用して GitHub から簡単に複製できます。その他の要件は、ncc-establish-subscription.py ツールで機能する、パッチが適用された nclient ツールを使用するこ

とです。これは、[ncc GitHub] ページの指示に従うか、または次の 2 つのコマンドを実行することでインストールできます。

```
$ git clone https://github.com/CiscoDevNet/ncc
$ sudo pip install --upgrade git+https://github.com/CiscoDevNet/ncclient.git
```

ダウンロードされたら、サブスクリプションは次のコマンドを実行することで確立できます。

```
$ python ncc-establish-subscription.py --host 10.10.10.223 --port 830 -x
"/wireless-access-point-oper:access-point-oper-data/wireless-interface-data" --period 5000 -u netconf -p netconf
```

```
python ncc-establish-subscription.py --host 10.10.10.223 --port 830 -x "/wireless-access-point-oper:access-point-oper-data/wireless-interface-data" --period 1000 -u netconf -p netconf
No handlers could be found for logger "ncclient.operations.subscribe"
Subscription Result : notif-bis:ok
Subscription Id      : 2147483655
-->>
(Default Callback)
Event time          : 2018-10-25 21:52:55.450000+00:00
Subscription Id    : 2147483655
Type               : 1
Data              :
<datastore-contents-xml xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <access-point-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper">
    <wireless-interface-data>
      <intf-name>GigabitEthernet1</intf-name>
      <intf-type>Management</intf-type>
      <intf-id>0</intf-id>
      <mgmt-ip>10.10.10.223</mgmt-ip>
      <net-mask>255.255.255.0</net-mask>
      <mgmt-mac>52:54:00:02:35:FA</mgmt-mac>
    </wireless-interface-data>
  </access-point-oper-data>
</datastore-contents-xml>
```

上の例では、「/wireless-access-point-oper:access-point-oper-data/wireless-interface-data」の xpath を持つ Cisco-IOS-XE-wireless-access-point-oper YANG モデルを使用してテレメトリ サブスクリプションが作成されています。次のスクリーンショットに示すように、YangExplorer を使用してこの YANG モデル用の適切な XPath フィルタを決定します。

The screenshot shows the YangExplorer web interface. On the left, the 'Explorer' pane shows a tree view of YANG models, with 'Cisco-IOS-XE-wireless-access-point-oper' selected and expanded to 'wireless-interface-data'. The main area displays configuration options for a device profile, including 'Profile' (cat9800), 'Platform' (other), 'Host' (10.10.10.223), 'Port' (830), 'Username' (netconf), and 'Password' (netconf). Below this, the 'Encoding' section shows a RESTCONF message with an XPath filter: `<filter><access-point-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper"><wireless-interface-data/></access-point-oper-data></filter>`. On the right, the 'Property Value' table lists details for the selected node, with the 'XPath Filter' property highlighted in red, showing the path: `/wireless-access-point-oper:access-point-oper-data/wireless-interface-data`.

## ダイヤルアウト gRPC 設定済みサブスクリプション

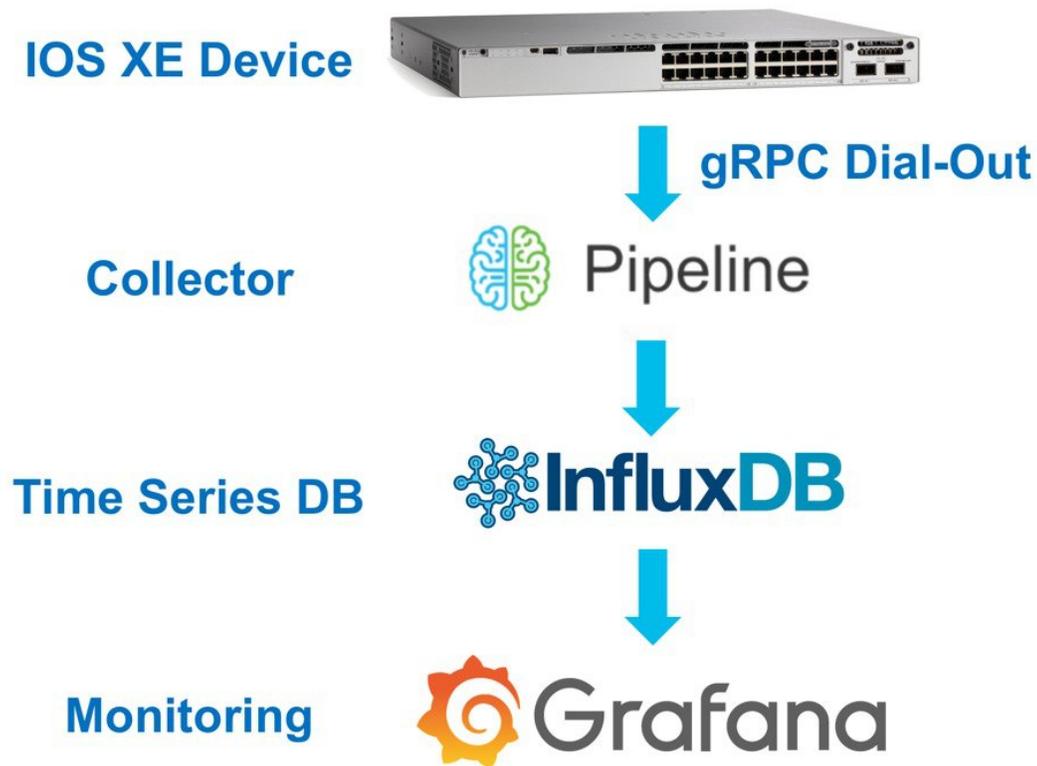
サブスクライバまたはコレクタでデバイスへのテレメトリ サブスクリプションを確立するのではなく、ダイヤルアウト サブスクリプションでデバイスを設定できます。このモードでのコレクタのロールは、データを収集して処理することだけです。テレメトリデータを受信するためにデバイスへのセッションを確立することはロールに含まれません。ダイヤルアウト サブスクリプションのみが gRPC-TCP 接続を介したキー値 Google プロトコルバッファ (kv-gp) でエンコーディングをサポートします。

次の設定を使用してダイヤルアウト gRPC テレメトリ サブスクリプションを確立できます。この設定は、101 という ID で新しいサブスクリプションを作成します。エンコーディングは kv-gpb に設定され、xpath フィルタが KPI を定義して、この場合はワイヤレスインターフェイス データをサブスクライブします。xpath フィルタは YANG モデル内で定義されます。YangExplorer を使用してこのモデルの正確な xpath を決定します。デバイスから使用する送信元アドレスと VRF が設定され、受信者 IP、ポートおよびプロトコルも含まれます。yang-push ストリームは 1000 センチ秒に設定されるため、データは 10 秒ごとに公開されます。

```
telemetry ietf subscription 101
  encoding encode-kvgpb
  filter xpath /wireless-access-point-oper:access-point-oper-data/wireless-interface-data
  source-address 10.10.10.223
  source-vrf Mgmt-vrf
  stream yang-push
  update-policy periodic 1000
  receiver ip address 10.10.10.49 57500 protocol grpc-tcp
```

## パイプラインを使用したイベントの受信

gRPC インターフェイスを介して送信されるキー値暗号化 Google プロトコルバッファ (kv-gpb) テレメトリ データは、ビジネス ニーズと使用例に応じて多くのツールおよびさまざまな設定で受信できます。パイプラインはデータの受信に使用できるツールの一連のオープンソースであり、<https://github.com/cisco/bigmuddy-network-telemetry-pipeline> の GitHub で入手できます。パイプラインは gRPC レシーバとして機能するように動作し、Google プロトコルバッファを処理してデータを時系列データベース InfluxDB に送信します。そこから、Grafana を使用してデータを可視化できます。パイプラインは高度に設定可能です。さまざまなテレメトリ ソースを受信できるだけでなく、Kafka、InfluxDB、Prometheus などのさまざまなデータ ソースにデータを出力できます。[GitHub] ページのドキュメントに従ってパイプラインをセットアップし、必要に応じてテレメトリを受信します。からその他のドキュメントを入手できます。 <https://xrdocs.io/telemetry/tutorials/2018-06-04-ios-xr-telemetry-collection-stack-intro/>



## テレメトリ サブスクリプションの確認

テレメトリ サブスクリプション設定のステータスを確認するために使用できるいくつかの `show` コマンドがあります。

```
show telemetry ietf subscription all
show telemetry ietf subscription <ID> receiver
show telemetry ietf subscription all <ID> detail
```

次にそれぞれの例を示します。

```
WLC#show telemetry ietf subscription all
Telemetry subscription brief
```

ID	Type	State	Filter type
2147483653	Dynamic	Valid	xpath

```
WLC#show telemetry ietf subscription 2147483654 detail
Telemetry subscription detail:

Subscription ID: 2147483654
Type: Dynamic
State: Valid
Stream: yang-push
Filter:
  Filter type: xpath
  XPath: /wireless-access-point-oper:access-point-oper-data/wireless-access-point-oper-data
Update policy:
  Update Trigger: periodic
  Period: 1000
Encoding: encode-xml
Source VRF:
Source Address:
Notes:

Receivers:
  Address      Port      Protocol      Protocol Prof
-----
10.10.10.49   43824     netconf
```

## まとめ

このガイド内の詳細情報により、使用するプログラミング対応インターフェイスを有効にして設定することができます。設定済みテレメトリサブスクリプションおよびダイナミックテレメトリサブスクリプションは、オープンソースツールを使用して確立できます。YangExplorer ツールを使用して WLAN をプログラムで作成、確認、削除するには、XML ペイロードの例を使用できます。Ansible 設定の例を使用すれば、Catalyst 9800 デバイス上でプログラミング対応インターフェイスを有効にすることができます。

## 関連リソース

Cisco IOS-XE Programmability Book <https://www.cisco.com/c/dam/en/us/products/collateral/enterprise-networks/nb-06-ios-xe-prog-ebook-cte-en.pdf>

# Cisco IOS XE Programmability Automating Device Lifecycle Management



## Programmability Configuration Guide

[https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/169/b\\_169\\_programmability\\_cg.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/169/b_169_programmability_cg.html)

## 参照先

- YangExplorer : <https://github.com/CiscoDevNet/yang-explorer>
- pYang : <https://github.com/mbj4668/pyang>
- DevNet のパイプライン : <https://developer.cisco.com/codeexchange/github/repo/cisco/bigmuddy-network-telemetry-pipeline/>

- GitHub のパイプライン : <https://github.com/cisco/bigmuddy-network-telemetry-pipeline>
- XR のパイプラインのドキュメント : <https://xrdocs.io/telemetry/tutorials/2018-06-04-ios-xr-telemetry-collection-stack-intro/>
- NCC : <https://github.com/CiscoDevNet/ncc>

## ご質問がある場合

- Cisco DevNet: <https://developer.cisco.com/>
- シスコ コミュニティ: <https://community.cisco.com/>

**【注意】** シスコ製品をご使用になる前に、安全上の注意（[www.cisco.com/jp/go/safety\\_warning/](http://www.cisco.com/jp/go/safety_warning/)）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

©2008 Cisco Systems, Inc. All rights reserved.

Cisco、Cisco Systems、およびCisco Systemsロゴは、Cisco Systems, Inc. またはその関連会社の米国およびその他の一定の国における登録商標または商標です。本書類またはウェブサイトに掲載されているその他の商標はそれぞれの権利者の財産です。

「パートナー」または「partner」という用語の使用はCiscoと他社との間のパートナーシップ関係を意味するものではありません。(0809R)

この資料の記載内容は2008年10月現在のものです。

この資料に記載された仕様は予告なく変更する場合があります。



#### 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/>