Network Monitoring

Network Monitoring

The mechanism that is used to transfer data to the third-party system is NETCONF/YANG. YANG can be used with the Network Configuration Protocol (NETCONF) to provide the desired solution of automated and programmable network operations.

You can contact the API or Developer Support for NETCONF/YANG features using the following link:

https://developer.cisco.com/site/support/#

The two types of information provided are:

- Status information received synchronously - NETCONF is the management interface used for status information, which allows to publish the operational state of the device, including the controller .

- Alarm and event information sent asynchronously - NETCONF/YANG push is the solution used for alarm and event information, which provides the mechanism to send NETCONF notifications subscribed for.

Note   When using NETCONF, you may see a user named "yang_mgmt_infra" showing up on accounting records (TACACS/RADIUS). This user is internal; when you pull data from Cisco IOS, it is expected to see this user.

Status Information Received Synchronously - Configuration Examples

NETCONF/YANG interface is used to accomplish customer requests.

The prerequisite configuration for Status Information and Alarm and Event Information is to enable NETCONF server on the controller by using the following command:

netconf-yang

> 📝
>
> **Note** The Cisco Catalyst 9800 wireless controller currently only supports RSA keys for the trustpoint used by the ncsshd process. Using EC keys instead of the RSA keys will cause the ncsshd process to crash and it will prevent using NETCONF.

The above command not only enables notifications, but also allows for configuration and operation access (OAM) via Netconf/Yang. For more information on Netconf/Yang, see the *NETCONF Protocol* chapter of the Programmability Configuration Guide at: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/products-installation-and-configuration-guides-list.html

In the Status Information Received Synchronously type, the following information is exported though NETCONF:

- Name of the village

- APs in each village

- Status of each AP

- Number of clients currently connected and logged on in each village and each AP

All the data for the items listed above is already available as the controller operational data exported through NETCONF. The examples below explain where the data items listed are available.

The following command is used in the controller :

```
wireless tag site village_name_1
```

The site tags can be retrieved by NETCONF using the **get-config** operation.

Example output for **Name of the Village**:

```
<site-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-site-cfg">
[...]
<site-tag-configs>
  <site-tag-config>
    <site-tag-name>village_name_1</site-tag-name>
    <description>custom user site tag for a village</description>
  </site-tag-config>
[...]
</site-tag-configs>
```

The controller 's operational data contains all the connected (joined) APs and lists their site tags. The example output displays the detailed information about the APs and the site tags. The following example displays the relevant fields and the corresponding controller show commands:

Example output of **Access Point per Village**:

```
<data>
    <access-point-oper-data
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper">
    [...]
      <radio-oper-data>
        <wtp-mac>00:1b:0c:00:02:00</wtp-mac>   #show ap dot11 {24ghz|5ghz} summary "MAC
Address"
        <radio-slot-id>0</radio-slot-id>        #show ap dot11 {24ghz|5ghz} summary "Slot"
        <ap-mac>00:1b:0c:00:02:00</ap-mac>
        <slot-id>0</slot-id>
```

```
            <radio-type>1</radio-type>               # 1 - 2.4GHz, 2 - 5GHz
            <admin-state>enabled</admin-state>       #show ap dot11 {24ghz|5ghz} summary "Admin
State"
            <oper-state>radio-up</oper-state>        #show ap dot11 {24ghz|5ghz} summary "Oper
State"
      [...]
[...]
      <capwap-data>
            <wtp-mac>00:1b:0c:00:02:00</wtp-mac>                   #show ap summary  "Radio MAC"
            <ap-operation-state>registered</ap-operation-state> #show ap summary  "State"
            <ip-addr>10.102.140.10</ip-addr>                       #show ap summary  "IP Address"
      [...]
            <admin-state>1</admin-state>                 #show ap status   "Status", 1 - Enabled,
 2 - Disabled
            <location>default-location </location>     #show ap summary  "Location"
            <country-code>CH </country-code>
            <name>AP_A-1</name>                          #show ap summary  "AP Name"
 [...]
            <tag-info>
              [...]
              <site-tag>
              <site-tag-name>village_name_1</site-tag-name> #show ap name AP_A-1 config general
 "Site Tag Name"
                 [...]
              </site-tag>
[...]
```

The operational data of the controller contains all the connected wireless clients information, which includes detailed client device information, such as the MAC address, IP address, State and the AP name.

Example output of the **Number of clients currently online and logged in each village and each AP**:

```
<data>
    <client-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-client-oper">
      <common-oper-data>
        <client-mac>00:00:1a:04:00:02</client-mac>      #show wireless client summary  "MAC
 Address"
        <ap-name>AP_A-1</ap-name>                           #show wireless client summary  "AP
Name"
        [...]
      <co-state>client-status-run</co-state>         #show wireless client summary  "State"
```

Alarm and Event Information Received Asynchronously - Configuration Examples

The push functionality for the alarm and event information is fulfilled with on-change notifications through NETCONF dynamic subscriptions, with XML encoding.

Example output of **AP Up/Down Events - Subscription**

```
Request:

<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="urn:uuid:b0c581c9-ff5a-4352-9e64-7f2ce1ec603a"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
```

```
  xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
 <stream>yp:yang-push</stream>
 <yp:xpath-filter>/access-point-oper-data/capwap-data/ap-operation-state</yp:xpath-filter>

 <yp:dampening-period>0</yp:dampening-period>
 </establish-subscription>
</rpc>

Reply:

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="urn:uuid:673b42b2-e988-4e20-a6c3-0679c08e6114"><subscription-result
xmlns='urn:ietf:params:xml:ns:yang:ietf-event-notifications'
xmlns:notif-bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:ok</subscription-result>
<subscription-id
xmlns='urn:ietf:params:xml:ns:yang:ietf-event-notifications'>2147483652</subscription-id>
</rpc-reply>
-->>
(Default Callback)
Event time      : 2018-03-09 15:08:21.880000+00:00
Subscription Id : 2147483651
Type            : 2
Data            :
<datastore-changes-xml xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
    <patch-id>null</patch-id>
    <edit>
      <edit-id>edit1</edit-id>
      <operation>merge</operation>
      <target>/access-point-oper-data/capwap-data</target>
      <value>
       <capwap-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-access-point-oper">

          <ap-operation-state>registered</ap-operation-state>
          <wtp-mac>00ab11006600</wtp-mac>
        </capwap-data>
      </value>
    </edit>
  </yang-patch>
</datastore-changes-xml>
<<--
```