



New Features for Cisco IOS-XE 17.5.1

The following are the new features available on the IR1101 for IOS-XE release 17.5.1:

- [DSL SFP Annex J support, on page 1](#)
- [VXLAN, on page 2](#)
- [Dying-Gasp SMS Notification for EM74XX Modems, on page 3](#)
- [SNMP MIB for Digital I/O, on page 4](#)
- [GPS access to IOx Apps, on page 4](#)
- [Yang model for mSATA, on page 5](#)
- [Guest Shell as IOx Container APP, on page 6](#)
- [SNMP MIB supports the show power CLI, on page 7](#)
- [ERSPAN Support Cellular Interface as Source Interface, on page 8](#)
- [Yang Model for DSL, on page 8](#)
- [DNP3 Enhancement, on page 9](#)

DSL SFP Annex J support

IOS-XE release 17.5.1 adds in support for Annex-J configuration in the controller interface.



Note ADSL2+ J is supported, ADSL2 J is not yet supported in 17.5.1.

To enable Annex-J, perform the following:

```
router#config term
router(conf)#controller vdsl 0/0/0
router(conf-if)#capability annex-j
router#(conf-if)#exit
router#
```

To remove Annex-J, perform the following:

```
To remove Annex-J:
router#config term
router(conf)#controller vdsl 0/0/0
router(conf-if)#no capability annex-j
```

```
router#(conf-if)#exit
router#
```

17.5.1 adds in a new command **rx-padding**. This command is used for packets with an MTU less than 64 bytes.



Important If frames less than 64mtu are expected downstream from the service provider, the Vlan configuration must be vlan 96.

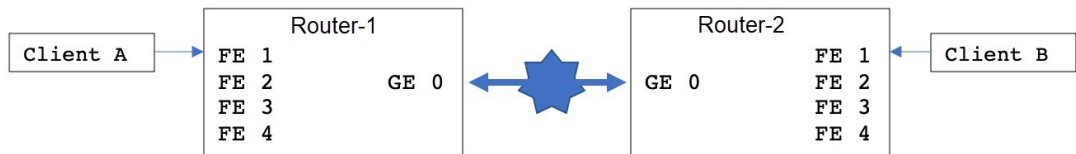
The command example is as follows:

```
router#config term
router#controller vds1 0/0/0
router(conf-if)#rx-padding
router(conf-if)#end
```

Execute **write mem** to save the configuration.

VXLAN

VXLAN is a MAC in IP/UDP (MAC-in-UDP) encapsulation technique with a 24-bit segment identifier in the form of a VXLAN ID. The larger VXLAN ID allows LAN segments to scale to 16 million in a cloud network. In addition, the IP/UDP encapsulation allows each LAN segment to be extended across existing Layer 3 networks, making use of Layer 3 Equal-Cost Multi-Path (ECMP).



The configuration for the two devices is shown in the following table:

Router-1	Router-2
<pre> bridge-domain 1 member vni 6001 member Vlan100 service-instance 1 ! interface Loopback1 ip address 200.200.200.200 255.255.255.255 ! interface GigabitEthernet0/0/0 ip address 192.168.1.2 255.255.255.0 media-type rj45 ! interface FastEthernet0/0/1 switchport access vlan 100 ! interface Vlan100 no ip address service instance 1 ethernet encapsulation dot1q 100 //untag ! interface nve1 no ip address source-interface Loopback1 member vni 6001 ingress-replication 100.100.100.100 ! ip forward-protocol nd ip pim rp-address 200.200.200.200 ip http server ip http secure-server ip route 0.0.0.0 0.0.0.0 192.168.1.3 ! </pre>	<pre> bridge-domain 1 member vni 6001 member Vlan100 service-instance 1 ! interface Loopback1 ip address 100.100.100.100 255.255.255.255 ! interface GigabitEthernet0/0/0 ip address 192.168.1.3 255.255.255.0 media-type rj45 ! interface FastEthernet0/0/1 switchport access vlan 100 ! interface Vlan100 no ip address service instance 1 ethernet encapsulation dot1q 100 //untag ! interface nve1 no ip address source-interface Loopback1 member vni 6001 ingress-replication 200.200.200.200 ! ip forward-protocol nd ip pim rp-address 100.100.100.100 no ip http server ip http secure-server ip route 0.0.0.0 0.0.0.0 192.168.1.2 ! </pre>

Dying-Gasp SMS Notification for EM74XX Modems

Prerequisites:

- Hardware Peripheral: P-LTEA-EA, P-LTEA-LA
- Initial Release: IOS-XE 17.5.1
- License: Cisco Network-advantage

Pluggable Interface Modules (PIMs) using the EM7430 or EM7455 modem have extra capacitors to supply power to the modem in case of loss of power to the module. This allows a graceful power off of the modem. When loss of power is detected, the modem is expected to send out dying gasp SMS when configured.

The following is an example of configuring dying gasp with a phone number and SMS message:

```

#controller Cellular 0/1/0
#lte dyinggasp sms send 9119110911 "Losing Power"
Warning: Enabling Dying Gasp SMS configuration completed successfully.
Please reset Modem for the changes to take effect

```

Configuration Steps

Step	Command	Purpose
1	configure terminal	Enters the global configuration mode.
2	controller Cellular <slot>	Enters the interface command mode for the cellular module controller slot.
3	lte dyinggasp detach enable	Enable dying-gasp feature with send detach request
4	lte dyinggasp sms send <phone number> <SMS message>	Configure the phone number to receive SMS text message and the content of text message to be sent by the modem when platform or module powered down.
5	exit	Exit configuration
6	write mem	Save changes to the router configuration

Configuration Example

The following example shows how to enable dying-gasp feature on cellular module in slot 0/1/0, specify phone number receiving the SMS, and the specific SMS text message to be sent by modem upon power failure.

```
router# configure terminal

router(config)# controller cellular 0/1/0
router (config-controller)# lte dyinggasp detach enable
router (config-controller)# lte dyinggasp sms send 4081112222
IR1101-#999_EM7455_powered_off!
```

SNMP MIB for Digital I/O

Digital I/O is similar to the ALARM IN and ALARM OUT supported in other IR devices. On other devices, ALARM IN is a dedicated input and the ALARM OUT is a dedicated output. With Digital I/O it can be input or output. There are 4 Digital I/O available on the IR1101 with an IRM-1100 Expansion Module.

MIB support will reflect the show alarm output for digital I/O only.

CISCO-DIGITAL-IO-MIB.my will have 4 digital I/O nodes. Each digital I/O node will have corresponding attributes like description, enable, severity, application, output, threshold, trigger leaf nodes for each digital I/O nodes.

GPS access to IOx Apps

Previously, when a modem has GPS enabled, the NMEA stream was not forwarded to IOx. This release allows the NMEA stream to be forwarded to IOx from the ngiolite module. There are two steps to enable this.

- Create a tunnel between Linux and IOx
- Forward all NMEA messages over the tunnel to IOx.

The system code checks for the presence of the tunnel, and if it is not present, data cannot be sent to IOx.

To support this feature there will be two new tunnels created for two cellular modems on the IR1101 and IR1800. Two tunnels are created by default and whichever modem has the GPS/NMEA enabled, the NMEA stream will be sent over the corresponding tunnel as follows:

Modem0:

[Linux] /dev/ttyTun5 and /dev/ttyTun6 [IOx]. Soft link to /dev/ttyTun5 will be created named /dev/ttyTunNMEA0, soft link to /dev/ttyTun6 will be created named /dev/ttyNMEA0 which can be accessed from IOx.

Modem1:

[Linux] /dev/ttyTun7 and /dev/ttyTun8 [IOx]. Soft link to /dev/ttyTun7 will be created named /dev/ttyTunNMEA1, soft link to /dev/ttyTun8 will be created named /dev/ttyNMEA1 which can be accessed from IOx.

The following command shows the state of the GPS:

```
IR1101#show app-hosting list
App id State
-----
gps RUNNING
```

Yang model for mSATA

YANG is a popular data modeling language to represent data sent over network management protocols such as NETCONF and RESTCONF. The Cisco-IOS-XE-device-hardware-oper YANG model has been modified to show mSATA information. mSATA has two CLIs to display associated data.

These two CLIs are:

show platform hardware msata status

- The CLI gives information on whether the SSD is present or not.
- If the SSD is present, a message "SSD is present" is displayed.
- If the SSD is not present, a message "SSD is not present" is displayed.

show platform hardware msata lifetime

- If SSD is present an output representing the SSD lifetime in % is displayed: "SSD lifetime remaining (%): 99"
- If SSD is not present, a message "SSD is not present" is displayed.

A typical YANG response for mSATA in device-inventory is as shown below:

```
<device-inventory>
  <hw-type>hw-type-ssd</hw-type>
  <hw-dev-index>5</hw-dev-index>
  <version>V00</version>
  <part-number>IR-SSD-MSATA-100G</part-number>
  <serial-number>FOC21520XFV</serial-number>
  <hw-description>mSATA Module</hw-description>
  <dev-name>Expansion module 2 - mSATA Module</dev-name>
```

```

    <field-replaceable>true</field-replaceable>
    <hw-class>hw-class-virtual</hw-class>
    <lifetime>99</lifetime>
  </device-inventory>

```

Cisco IOS-XE Yang Data Models are found here:

<https://github.com/YangModels/yang/tree/master/vendor/cisco/xe>

Each release has a directory, and the 17.5.1 release is found under 1751.

Guest Shell as IOx Container APP

The Guest Shell is a virtualized Linux-based environment, designed to run custom Linux applications, including Python for automated control and management of Cisco devices. Using the Guest Shell, the user can also install, update, and operate third-party Linux applications and access the IOS CLI.

The Guest Shell environment is intended for tools, Linux utilities, and manageability rather than networking.

Guest Shell shares the kernel with the host (router) system. Users can access the Linux shell of Guest Shell and update scripts and software packages in the container rootfs. However, users within the Guest Shell cannot modify the host file system and processes.

The Guest Shell container is managed using IOx. IOx is Cisco's Application Hosting Infrastructure for Cisco IOS XE devices. IOx enables hosting of applications and services developed by Cisco, partners, and third-party developers in network edge devices, seamlessly across diverse and disparate hardware platforms.

The Guest Shell is typically bundled with the system image and can be installed using the **guestshell enable** Cisco IOS command. However, this approach leads to an increase of roughly 75MB in the size of the image. This is a problem for some users who have limited bandwidth, or download images through LTE.

With these users in mind, guestshell will be made available as a single tar file which can then be downloaded and installed on the system like any other IOX application. As a result, there won't be any increase in the size of the universal release image.



Note Day 0 guestshell provisioning will not work with this approach.

By default, Guest Shell allows applications to access the management network via the management interface. For platforms like the IR1101, which don't have a dedicated management port, a VirtualPortGroup can be associated with Guest Shell in the IOS configuration.

Sample guestshell configuration can be found [here](#).

To install guestshell on the device, copy the tar file to the router and run the following command:

```
app-hosting install appid guestshell package <path to tar file>
```

Use the following command to check the status:

```
show app-hosting list
```

Once guestshell has been deployed successfully, standard guestshell commands such as **guestshell enable**, **guestshell run bash**, and **guestshell run python3** should work.

The following resource talks about running python scripts using guestshell:

[CLI Python Module](#)



Note Only python3 is supported in 17.5.1.

Important - Before You Install

Before attempting to install Guest shell on your device, please verify that the device has IOx container keys programmed on it by running the following command:

```
Router#show software authenticity keys | i Name
Product Name : SFP-VADSL2-I
Product Name : SFP-VADSL2-I
Product Name : IR1101
Product Name : IR1101
Product Name : Cisco Services Containers
Product Name : Cisco Services Containers
```

The output should contain one or more lines with the Product Name “Cisco Services Containers”. If the device doesn’t have container keys programmed on it, then you won’t be able to install guest shell.

You will see an error like the following:

```
*Aug 26 15:47:21.484: %IOSXE-3-PLATFORM: R0/0: IOx: App signature verification failed with
non-zero exit code
*Aug 26 15:47:21.588: %IM-6-INSTALL_MSG: R0/0: ioxman: app-hosting: Install failed: App
package signature (package.sign)
verification failed for package manifest file package.mf. Re-sign the application and then
deploy again.
```

There is no software based mechanism to install container keys on the device. The keys have to be programmed at the manufacturing facility. IR1100 devices shipped after January 1, 2020, should have the container keys programmed.

The guest shell tar file is published along with the IOS-XE image for a given release. More information can be found here:<https://developer.cisco.com/docs/iox/#!/iox-resource-downloads/downloads>

SNMP MIB supports the show power CLI

SNMP MIB support for the **show power** CLI is available through a new mib file: CISCO-ENTITY-SENSOR-MIB.my

The following is an example of the **show power** CLI:

```
#show power
Main PSU :
  Total Power Consumed: 8.77 Watts
  Configured Mode : N/A
  Current runtime state same : N/A
  PowerSupplySource : External PS
```

The following is an example of the CISCO-ENTITY-SENSOR-MIB.my MIB:

```
SensorDataType (INTEGER) watts(6)
SensorDataScale (INTEGER) milli(8)
SensorValue(INTEGER) 8770
```

Use the following commands to configure:

```
Router#config term
Router#(config) snmp-server community public RW
Router#(config) end
```

ERSPAN Support Cellular Interface as Source Interface

Encapsulated Remote Switched Port Analyzer (ERSPAN) allows traffic from Cellular interfaces to be monitored. ERSPAN sends monitored traffic to a network analyzer.

The following is a sample configuration:

```
Router(config)#monitor session 1 type erspan-source
Router(config-mon-erspan-src)#no shut
Router(config-mon-erspan-src)#source interface Cellular0/1/0
Router(config-mon-erspan-src)#destination
Router(config-mon-erspan-src-dst)#erspan-id 1
Router(config-mon-erspan-src-dst)#mtu 146
Router(config-mon-erspan-src-dst)#ip address 169.254.1.2
Router(config-mon-erspan-src-dst)#origin ip address 169.254.1.1
Router#show monitor session erspan-source
Session 1
-----
Type : ERSPAN Source Session
Status : Admin Enabled
Source Ports :
Both : Ce0/1/0
Destination IP Address : 169.254.1.2
MTU : 1464
Destination ERSPAN ID : 1
Origin IP Address : 169.254.1.1
```

For detailed information on configuring ERSPAN, see [Configuring ERSPAN](#).

Yang Model for DSL

YANG is a popular data modeling language to represent data sent over network management protocols such as NETCONF and RESTCONF.

The **Cisco-IOS-XE-controller-vdsl-oper** has been introduced to edit the Controller vdsl configurations which gives the yang support for the DSL.

An example of a typical yang response for edit config of the dsl controller follows:

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <controller>
    <VDSL xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-controller">
      <name>0/0/0</name>
      <adsl-pvc xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-adsl">
        <vpi-vci>255/65535</vpi-vci>
        <bridge-dot1q>21</bridge-dot1q>
        <encapsulation>vcmux</encapsulation>
      </adsl-pvc>
    </VDSL>
  </controller>
</native>
</nc:config></nc:edit-config></nc:rpc>
```




Note The Controller configurations can be retrieved using **get** and **get-config** operations with the Cisco-IOS-XE-native yang model.

Cisco IOS-XE Yang Data Models are found here:

<https://github.com/YangModels/yang/tree/master/vendor/cisco/xe>

Each release has a directory, and the 17.5.1 release is found under 1751.

DNP3 Enhancement

In some cases, older RTUs were previously used in peer-to-peer mode. These RTUs dynamically swapped the roles of DNP3 Serial subordinate and primary by setting the bit DIR=1 in the message header. ASE's SCADA stack used in Cisco routers are always configured to be DNP3 Serial primary. In this case, all the packets received from DNP3 serial with DIR=1 were ignored causing many messages from RTU to be discarded. To handle these scenarios, a new SCADA configuration CLI has been added:

scada-gw protocol ignore direction.

Enabling this CLI will allow the router to accept incoming packets from RTU even when DIR=1. The new CLI will also be added to the Cisco-IOS-XE-scada-gw.yang config model.

The following is an example usage:

```
Router# config term
Router(config)# scada-gw protocol ignore direction
```

Configuration Example

Configuration example with scada-gw protocol ignore direction on T101/T104

```
scada-gw protocol t101
channel rt-chan
link-addr-size two
bind-to-interface Async0/2/0
session rt-sess
attach-to-channel rt-chan
common-addr-size one
cot-size two
info-obj-addr-size three
link-addr 31
sector rt-sec
attach-to-session rt-sess
asdu-addr 100
scada-gw protocol t104
channel mt-chan
t3-timeout 20
tcp-connection 0 local-port 8001 remote-ip 192.168.1.0/24
session mt-sess
attach-to-channel mt-chan
sector mt-sec
attach-to-session mt-sess
asdu-addr 101
map-to-sector rt-sec
```

```
scada-gw protocol ignore direction  
scada-gw enable
```