



## Manage Collection Jobs

---

This section contains the following topics:

- [About Collection Jobs, on page 1](#)
- [Create a Collection Job, on page 30](#)
- [Delete a Collection Job, on page 35](#)
- [Monitoring Collection Jobs, on page 35](#)
- [List of Pre-loaded Traps and MIBs for SNMP Collection, on page 39](#)
- [List of Pre-loaded YANG Modules for MDT Collection, on page 45](#)

## About Collection Jobs

A collection job describes what task a Cisco Crosswork Data Gateway is expected to perform. Applications request data collection via collection jobs. Cisco Crosswork then assigns these collection jobs to a Cisco Crosswork Data Gateway to serve the request.

You can collect more than one type of data at a time by using separate collection jobs.

For each collection job you create, Cisco Crosswork Data Gateway executes the collection request and deposits the collected data in the preferred data destination(s).

Cisco Crosswork Data Gateway lets you create following types of collection jobs:

- [CLI Collection Job, on page 2](#)
- [SNMP Collection Job, on page 3](#)
- [MDT Collection Job, on page 11](#)
- [gNMI Collection Job, on page 12](#)
- [Syslog Collection Job, on page 20](#)



---

**Note**

1. Cisco Crosswork Data Gateway drops incoming traffic if there is no corresponding (listening) collection job request for the same. It also drops data, syslog events and SNMP traps received from an unsolicited device (i.e., not attached to Crosswork Data Gateway).
  2. Polled data cannot be requested from the device until Cisco Crosswork Data Gateway is ready to process and transmit the data.
-

### Smart Licensing for Active Collection Jobs

To be able to create collection jobs that can forward data to third-party destinations, ensure that the following Smart Licensing requirements are met:

1. From the main menu, go to **Administration > Application Management > Smart License** and select the Cisco Crosswork application.
2. Ensure that the status is as follows:
  - **Registration Status - Registered**  
Indicates that you have registered with Cisco Smart Software Manager (CSSM) and are authorized to use the reserved licensed features.
  - **License Authorization Status - Authorized (In Compliance)**  
Indicates you have not exceeded the device count in the external collection jobs

In the Evaluation period (**Registration Status** is Unregistered and the **License Authorization Status** is **Evaluation mode**), you will be able to create collection jobs until the evaluation period expires. After this, you must register with Cisco Smart Software Manager (CSSM) to use licensed features. See Section: [Manage Licenses](#) for more information.

If you do not register with Cisco Smart Software Manager (CSSM) after the Evaluation period has expired, you will not be able to create collection jobs. However, you can still view and delete any collection jobs.

## CLI Collection Job

Cisco Crosswork Data Gateway supports CLI-based data collection from the network devices. Only show commands are supported for this type of collection job.



### Note

- The initial status for all the collection jobs in the UI is Unknown. Upon receiving a CLI collection job, Cisco Crosswork Data Gateway performs basic validations on it. If the collection job is valid, its status changes to Successful, else it changes to Failed.
- Device should not have any banner configuration for CLI collection to work properly. Please refer to device documentation on how to turn this off.
- The value of **Cadence** is in seconds. It should be set either to 0 to indicate the sensor configured to be collected only once.  
OR  
It should be  $\geq 60$  (i.e. at least 1 minute) up to 2764800 seconds ( i.e. at most 32 days) max, indicating how frequently configured sensor data should be collected.
- When collection from a device is skipped due to previous execution still in progress, Cisco Crosswork Data Gateway raises a warning log. No alert is generated for this scenario.

Following is a sample of CLI collection job. For more information, see API documentation on [Cisco DevNet](#).

```
{
  "collection_job": {
    "application_context": {
```

```

    "context_id": "collection-job1",
    "application_id": "APP1"
  },
  "collection_mode": {
    "lifetime_type": "APPLICATION_MANAGED",
    "collector_type": "CLI_COLLECTOR"
  },
  "job_device_set": {
    "device_set": {
      "devices": {
        "device_ids": [
          "658adb03-cc61-448d-972f-4fcec32cbfe8"
        ]
      }
    }
  },
  "sensor_input_configs": [
    {
      "sensor_data": {
        "cli_sensor": {
          "command": "show platform"
        }
      },
      "cadence_in_millisecc": "tel:60000"
    }
  ],
  "sensor_output_configs": [
    {
      "sensor_data": {
        "cli_sensor": {
          "command": "show platform"
        }
      },
      "destination": {
        "destination_id": "1e71f2fb-ea65-4242-8efa-e33cec71b369",
        "context_id": "topic1"
      }
    }
  ]
}

```

## SNMP Collection Job

Cisco Crosswork Data Gateway supports SNMP-based data collection based on the OIDs supported on the devices.

The SNMP collector makes a poll request to Cisco Crosswork to get its configuration profile (a list of MIB objects to collect and a list of devices to fetch from). It determines the corresponding OIDs by looking up the pre-packaged list of MIB modules or the custom list of MIB modules.



**Note** Cisco Crosswork Data Gateway enables SNMP polling on third party devices for standard MIBs already included in the system. Proprietary MIBs are required only if the collection request references MIB TABLE names or SCALAR names from a proprietary MIB. However, if the requests are OID-based, then MIBs are not required.

Once the OIDs are resolved, they are provided as input to the SNMP collectors.

The device packages can be imported into the Crosswork Data Gateway VM as described in Section [Add a Custom Software Package](#).

The following SNMP versions are supported:

- SNMPv1
- SNMPv2c
- SNMPv3

The table below lists supported privacy protocols and the value that needs to be given in the collection payload for SNMP and SNMP Trap collection jobs:

Protocol	SNMP Collection Payload	SNMP Trap Collection Payload
aes	AES	N/A
des56	DES	N/A
3des	3DES	N/A
aes 128	AES128	N/A
aes 192	AES192 or CiscoAES192(Cisco specific)	N/A
aes 256	AES256 or CiscoAES256(Cisco specific)	N/A



#### Note

- The initial status for all the collection jobs in the UI is Unknown. Upon receiving a SNMP collection job, Cisco Crosswork Data Gateway performs basic validations on it. If the collection job is valid, its status changes to Successful, else it changes to Failed.
- The value of **Cadence** is in seconds. It should be set either to 0 to indicate the sensor configured to be collected only once.  
OR  
It should be  $\geq 60$  (i.e. at least 1 minute) up to 2764800 seconds ( i.e. at most 32 days) max, indicating how frequently configured sensor data should be collected.
- When collection from a device is skipped due to previous execution still in progress, Cisco Crosswork Data Gateway raises a warning log. No alert is generated for this scenario.
- For SNMP v1/v2c, if the device details (such as host or community string) are incorrect in the payload, Cisco Crosswork Data Gateway ignores the traps received from the device and logs a WARN message.
- Only SNMPv1 and SNMPv2c versions are supported for SNMP traps
- In case of SNMP v3, if the device details (such as auth, priv, and security name details) are incorrect in the payload, Cisco Crosswork Data Gateway filters it out and hence, does not receive the trap. Thus, no WARN message is logged.

#### Sample Configurations on Device:

Table 1:

Version	Command	To...
V1	<pre>snmp-server group &lt;group_name&gt; v1  snmp-server user &lt;user_name&gt; &lt;group_name&gt; v1</pre>	Define the SNMP version, user/user group details.
	<pre>snmp-server host &lt;host_ip&gt; traps &lt;community_string&gt; udp-port 1062  For example,  snmp-server host a.b.c.d traps test udp-port 1062</pre>	Define the destination to which trap data must be forwarded.
	<pre>snmp-server traps snmp linkup snmp-server traps snmp linkdown</pre>	Enable traps to notify link status.
V2c	<pre>snmp-server group &lt;group_name&gt; v2c  snmp-server user &lt;user_name&gt; &lt;group_name&gt; v2c</pre>	Define the SNMP version, user/user group details.
	<pre>snmp-server host &lt;host_ip&gt; traps SNMP version &lt;community_string&gt; udp-port 1062  snmp-server host a.b.c.d traps version 2c v2test udp-port 1062</pre>	Define the destination to which trap data must be forwarded.
	<pre>snmp-server traps snmp linkup snmp-server traps snmp linkdown</pre>	Enable traps to notify link status.

Version	Command	To...
V3	<pre>snmp-server group &lt;group_name&gt; v3 auth notify &lt;user_name&gt; read &lt;user_name&gt; write &lt;user_name&gt;  snmp-server view &lt;user_name&gt; 1.3 included</pre>	Define the SNMP version, user/user group details.
	<pre>snmp-server user &lt;user_name&gt; &lt;group_name&gt; v3 auth md5 &lt;password&gt; priv aes 128 &lt;password&gt;  snmp-server host &lt;host_IP&gt; traps version 3 priv &lt;user_name&gt; udp-port 1062</pre>	Define the destination to which trap data must be forwarded.
	<pre>snmp-server traps snmp linkup  snmp-server traps snmp linkdown</pre>	Enable traps to notify link status.

The SNMP Collector supports the following operations:

- SCALAR
- TABLE
- MIB\_WALK
- TRAP
- DEVICE\_PACKAGE

These operations are defined in the sensor config (see payload sample below).



#### Note

There is an optional **deviceParams** attribute **snmpRequestTimeoutMillis** (not shown in the sample payloads) that should be used if the device response time is very high. It's not recommended to use **snmpRequestTimeoutMillis** unless you are absolutely certain that your device response time is very high.

The value for **snmpRequestTimeoutMillis** should be specified in milliseconds:

Default value is 1500 milliseconds

Minimum value is 1500 milliseconds

However, there is no limitation on the maximum value of this attribute.

Following is an SNMP collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    }
  }
}
```

```

},
"collection_mode": {
  "lifetime_type": "APPLICATION_MANAGED",
  "collector_type": "SNMP_COLLECTOR"
},
"job_device_set": {
  "device_set": {
    "devices": {
      "device_ids": [
        "c70fc034-0cbd-443f-ad3d-a30d4319f937",
        "8627c130-9127-4ed7-ace5-93d3b4321d5e",
        "c0067069-c8f6-4183-9e67-1f2e9bf56f58"
      ]
    }
  }
},
"sensor_input_configs": [
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.1.3.0",
          "snmp_operation": "SCALAR"
        }
      }
    },
    "cadence_in_millisecc": "60000"
  },
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.31.1.1",
          "snmp_operation": "TABLE"
        }
      }
    },
    "cadence_in_millisecc": "60000"
  }
],
"sensor_output_configs": [
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.1.3.0",
          "snmp_operation": "SCALAR"
        }
      }
    },
    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic1_461cb8aa-a16a-44b8-b79f-c3daf3ea925f"
    }
  },
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.31.1.1",
          "snmp_operation": "TABLE"
        }
      }
    }
  }
],

```

```

    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic2_e7ed6300-fc8c-47ee-8445-70e543057f8a"
    }
  ]
}
}
}

```

### SNMP Traps Collection Job

SNMP traps are handled in a similar manner. Trap listeners listen on a port and then dispatch data to recipients (based on their topic of interest).

Cisco Crosswork Data Gateway supports three types of non-yang/OID based traps:

sensor path	purpose
*	To get all the traps pushed from the device without any filter.
MIB level traps	OID of one MIB notifications (Ex: 1.3.6.1.2.1.138.0 to get all the isis-mib level traps)
Specific trap	OID of the specific trap (Ex: 1.3.6.1.6.3.1.1.5.4 to get the linkUp trap)



#### Note

- Device should have been pre-configured by the traps.
- Cisco Crosswork Data Gateway listens on UDP port 1062 for Traps.
- If the collection job is invalid, there is missing configuration on the device, or no trap is received, the status of the job remains "Unknown".
- For list of supported Traps and MIBs, see [List of Pre-loaded Traps and MIBs for SNMP Collection, on page 39](#)

On receiving a trap, Cisco Crosswork Data Gateway does the following validations:

1. Check if any collection job is created for the device.
2. Checks the trap version and community string.
3. For SNMP v3, validates for user auth and priv protocol and credentials.

Cisco Crosswork Data Gateway filters the traps based on the trap OID mentioned in the sensor path and sends only those requested.

Cisco Crosswork Data Gateway supports the following YANG paths:

sensor path	purpose
snmp-trap-raw-oper:traps/data	To get all the traps pushed from the device without any filter.
IF-MIB:notifications	To get all the IF-MIB notifications (ex: linkUp, linkDown, etc.)
ISIS-MIB:notifications	To get all the ISIS-MIB notifications.



sensor path	purpose
SNMPv2-MIB:notifications	To get all the snmpV2 Mib notifications.

Following is an SNMP-Trap collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "TRAP_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "a9b8f43d-130b-4866-a26a-4d0f9e07562a",
            "8c4431a0-f21d-452d-95a8-84323a19e0d6",
            "eaab2647-2351-40ae-bf94-6e4a3d79af3a"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "trap_sensor": {
            "path": "1.3.6.1.6.3.1.1.4"
          }
        },
        "cadence_in_millisec": "60000"
      }
    ],
    "sensor_output_configs": [
      {
        "sensor_data": {
          "trap_sensor": {
            "path": "1.3.6.1.6.3.1.1.4"
          }
        },
        "destination": {
          "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
          "context_id": "topic1_696600ae-80ee-4a02-96cb-3a01a2415324"
        }
      }
    ]
  }
}
```

### Enabling Traps forwarding to external applications

As per the current implementation, in case of an SNMP Trap collection job, all traps are sent to the specified data destination even if the SNMP Trap OID is not provided in the sensor path.



**Note** It is also recommended to selectively enable on the device only those traps that are needed by Crosswork.

To identify the type of trap from the data received on the destination, look for *oid* (OBJECT\_IDENTIFIER, for example, 1.3.6.1.6.3.1.1.4.1.0) and *strValue* associated to the *oid* in the *OidRecords* (application can match the OID of interest to determine the kind of trap).

Below are some sample values and a sample payload:

- Link up

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.6.3.1.1.5.4
```

- Link Down

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.6.3.1.1.5.3
```

- Syslog

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.4.1.9.9.41.2.0.1
```

- Cold Start

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.6.3.1.1.5.1
```

```
{
  "nodeIdStr": "BF5-XRV9K1.tr3.es",
  "nodeIdUuid": "C9tZ5lJoSJKf5OZ67+U5JQ==",
  "collectionId": "133",
  "collectionStartTime": "1580931985267",
  "msgTimestamp": "1580931985267",
  "dataGpbkv": [
    {
      "timestamp": "1580931985267",
      "name": "trapsensor.path",
      "snmpTrap": {
        "version": "V2c",
        "pduType": "TRAP",
        "v2v3Data": {
          "agentAddress": "172.70.39.227",
          "oidRecords": [
            {
              "oid": "1.3.6.1.2.1.1.3.0",
              "strValue": "7 days, 2:15:17.02"
            },
            {
              "oid": "1.3.6.1.6.3.1.1.4.1.0", // This oid is the Object Identifier.
              "strValue": "1.3.6.1.6.3.1.1.5.3" // This is the value that determines the
kind of trap.
            },
            {
              "oid": "1.3.6.1.2.1.2.2.1.1.8",
              "strValue": "8"
            },
            {
              "oid": "1.3.6.1.2.1.2.2.1.2.8",
              "strValue": "GigabitEthernet0/0/0/2"
            },
            {
              "oid": "1.3.6.1.2.1.2.2.1.3.8",
              "strValue": "6"
            },
            {
              "oid": "1.3.6.1.4.1.9.9.276.1.1.2.1.3.8",
              "strValue": "down"
            }
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
},
"collectionEndTime": "1580931985267",
"collectorUuid": "YmNjZjEzMTktZjF1OS00NTE5LWI4OTgtY2Y1ZmQxZDFjNWExO1RSQVBFQ09MTEVDVE9S",
"status": {
  "status": "SUCCESS"
},
"modelData": {},
"sensorData": {
  "trapSensor": {
    "path": "1.3.6.1.6.3.1.1.5.4"
  }
},
"applicationContexts": [
  {
    "applicationId": "APP1",
    "contextId": "collection-job-snmp-traps"
  }
]
}

```

## MDT Collection Job

Crosswork Data Gateway supports data collection from network devices using Model-driven Telemetry (MDT) to consume telemetry streams directly from devices (for IOS-XR based platforms only).



### Note

- MDT collector retains the collection ID that comes as part of the telemetry proto for the device. This behavior is different from CLI and SNMP collectors which compute the collection ID based on the sequence number of the collection.
- MDT collection jobs require some configuration to be done on the device. This configuration is automatically taken care of by NSO. Ensure that NSO is integrated and properly working
- If there is some change (delete/update) in existing MDT jobs between backup and restore operations, Cisco Crosswork does not replay the jobs for config update on the devices as it involves Provider(NSO). You have to restore configs on provider/devices. Cisco Crosswork will just restore the jobs in database.
- Before using any YANG modules, check if they are supported. See Section: [List of Pre-loaded YANG Modules for MDT Collection](#) , on page 45

It supports data collection for the following transport mode:

- MDT TCP Dial-out Mode

Following is a sample of MDT collection payload:

```

{
  "collection_job": {
    "job_device_set": {
      "device_set": {
        "device_group": "mdt"
      }
    },
    "sensor_output_configs": [{
      "sensor_data": {

```

```

    "mdt_sensor": {
      "path":
"Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters"

    }
  },
  "destination": {
    "context_id": "cw.mdt_sensor.cisco-ios-xr-infra-statsd-oper.gpb",
    "destination_id": "c2a8fba8-8363-3d22-b0c2-a9e449693fae"
  }
},
{
  "sensor_data": {
    "mdt_sensor": {
      "path": "Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate"

    }
  },
  "destination": {
    "context_id": "cw.mdt_sensor.cisco-ios-xr-infra-statsd-oper.gpb",
    "destination_id": "c2a8fba8-8363-3d22-b0c2-a9e449693fae"
  }
}
],
"sensor_input_configs": [{
  "sensor_data": {
    "mdt_sensor": {
      "path": "Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/data-rate"

    }
  },
  "cadence_in_millisec": "70000"
}, {
  "sensor_data": {
    "mdt_sensor": {
      "path":
"Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters"

    }
  },
  "cadence_in_millisec": "70000"
}
],
"application_context": {
  "context_id": "c4",
  "application_id": "a4-mdt"
},
"collection_mode": {
  "lifetime_type": "APPLICATION_MANAGED",
  "collector_type": "MDT_COLLECTOR"
}
}
}

```

## gNMI Collection Job

Cisco Crosswork supports gRPC Network Management Interface (gNMI) based telemetry data collection via Cisco Crosswork Data Gateway. It supports only gNMI Dial-In (gRPC Dial-In) streaming telemetry data based on subscription and relaying subsequent subscription response(notifications) to requested destinations.



**Note** gNMI collection is supported as long as the models are supported by the target device platform. gNMI must be configured on devices before you can submit gNMI collection jobs. Check platform-specific documentation.

For sample device configuration, see [Sample Device Configuration - gNMI, on page 14](#).

In gNMI, both secure and insecure mode can co-exist on the device. Cisco Crosswork gives preference to secure mode over insecure mode based on the information passed in the inventory.

If device reloads, gNMI collector ensures that the existing subscriptions are re-subscribed to the device.

gNMI specification does not have a way to mark end of message. Hence, Destination/Dispatch cadence is not supported in gNMI collector.

Cisco Crosswork Data Gateway supports all types of stream-based subscriptions:

- **SAMPLE** : Cadence-based collection.
- **ON\_CHANGE**: First response include the state of all the elements for the subscribed path, followed by subsequent updates to the changes leaf values.
- **TARGET\_DEFINED**: Router/Device choses the mode of subscription on a per-leaf basis based on the subscribed path (i.e. one of **SAMPLE** or **ON\_CHANGE**)



**Note**

- Cisco Crosswork Data Gateway relies on the device to declare the support of one or more modes.
- gNMI sensor path with default values does not appear payload. This is a known protobuf behavior.

For boolean default value will be false. For enum it is gnmi.proto specified.

Example 1:

```
message GNMIDeviceSetting {
  bool suppress_redundant = 1;
  bool allow_aggregation = 4;
  bool updates_only = 6;
}
```

Example 2:

```
enum SubscriptionMode {
  TARGET_DEFINED = 0; //default value will not be printed
  ON_CHANGE = 1;
  SAMPLE = 2;
}
```

Following is a sample gNMI collection payload:

```
{
  "collection_job": {
    "job_device_set": {
      "device_set": {
        "device_group": "gnmi"
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
```

```

        "gnmi_sensor": {
          "path": {
            "origin": "",
            "elem": [
              {
                "name": "interfaces"
              },
              {
                "name": "interface",
                "key": {
                  "name": "GigabitEthernet0/0/0/4"
                }
              }
            ]
          },
          "mode" : "SAMPLE"
        }
      },
      "cadence_in_millisec": "30000"
    }
  ],
  "sensor_output_configs": [
    {
      "sensor_data": {
        "gnmi_sensor": {
          "path": {
            "origin": "",
            "elem": [
              {
                "name": "interfaces"
              },
              {
                "name": "interface",
                "key": {
                  "name": "GigabitEthernet0/0/0/4"
                }
              }
            ]
          },
          "mode" : "SAMPLE"
        }
      },
      "destination": {
        "context_id": "topic_gnmi",
        "destination_id": "c2a8fba8-8363-3d22-b0c2-a9e449693fae"
      }
    }
  ],
  "application_context": {
    "context_id": "gnmi_test_context",
    "application_id": "gnmi"
  },
  "collection_mode": {
    "lifetime_type": "APPLICATION_MANAGED",
    "collector_type": "GNMI_COLLECTOR"
  }
}

```

## Sample Device Configuration - gNMI

### Cisco IOS XR devices

1. Enable gRPC over an HTTP/2 connection.

```
Router#configure
Router(config)#grpc
Router(config-grpc)#port <port-number>
```

The port number ranges from 57344 to 57999. If a port number is unavailable, an error is displayed.

## 2. Set the session parameters.

```
Router(config)#grpc{ address-family | dscp | max-request-per-user | max-request-total |
max-streams |
max-streams-per-user | no-tls | service-layer | tls-cipher | tls-mutual | tls-trustpoint
| vrf }
```

where:

- `address-family`: set the address family identifier type
- `dscp`: set QoS marking DSCP on transmitted gRPC
- `max-request-per-user`: set the maximum concurrent requests per user
- `max-request-total`: set the maximum concurrent requests in total
- `max-streams`: set the maximum number of concurrent gRPC requests. The maximum subscription limit is 128 requests. The default is 32 requests
- `max-streams-per-user`: set the maximum concurrent gRPC requests for each user. The maximum subscription limit is 128 requests. The default is 32 requests
- `no-tls`: disable transport layer security (TLS). The TLS is enabled by default.
- `service-layer`: enable the grpc service layer configuration
- `tls-cipher`: enable the gRPC TLS cipher suites
- `tls-mutual`: set the mutual authentication
- `tls-trustpoint`: configure trustpoint
- `server-vrf`: enable server vrf

## 3. Enable TPA (Traffic Protection for Third-Party Applications).

```
tpa
vrf default
address-family ipv4
default-route mgmt
update-source dataports MgmtEth0/RP0/CPU0/0
```

## Cisco IOS XE Devices

The following example shows how to enable the gNMI server in insecure mode:

```
Device# configure terminal
Device(config)# gnmi-yang
Device(config)# gnmi-yang server
Device(config)# gnmi-yang port 50000 <The default port is 50052.>
Device(config)# end
Device
```

The following example shows how to enable the gNMI server in secure mode:

Certs and trustpoint are only required for secure gNMI servers.

```

Device# configure terminal
Device(config)# gnmi-yang server
Device(config)# gnmi-yang secure-server
Device(config)# gnmi-yang secure-trustpoint trustpoint1
Device(config)# gnmi-yang secure-client-auth
Device(config)# gnmi-yang secure-port 50001 <The default port is 50051.>
Device(config)# end
Device

```

### Device certificates

Certs and trustpoint are only required for secure gNMI servers.

### Creating Certs with OpenSSL on Linux

The following example shows how to create Certs with OpenSSL on a Linux machine:

```

# Setting up a CA
openssl genrsa -out rootCA.key 2048
openssl req -subj /C=/ST=/L=/O=/CN=rootCA -x509 -new -nodes -key rootCA.key -sha256 -out
rootCA.pem

# Setting up device cert and key
openssl genrsa -out device.key 2048
openssl req -subj /C=/ST=/L=/O=/CN=<hostnameFQDN> -new -key device.key -out device.csr
openssl x509 -req -in device.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out
device.crt -sha256
# Encrypt device key - needed for input to IOS
openssl rsa -des3 -in device.key -out device.des3.key -passout pass:<password - remember
this for later>

# Setting up client cert and key
openssl genrsa -out client.key 2048
openssl req -subj /C=/ST=/L=/O=/CN=gnmi_client -new -key client.key -out client.csr
openssl x509 -req -in client.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out
client.crt -sha256

```

### Installing Certs on a Cisco IOS XR Device

To install certs on Cisco IOS XR, replace files in the following path:

1. Login into XR machine.
2. Type run command on terminal prompt.

```
RP/0/RP0/CPU0:xrvr-7.2.1#run
```

3. Navigate to the following directory:

```
cd /misc/config/grpc
```

4. Replace the content of the following files:
  - replace contents of ems.pem with device.crt
  - replace contents of ems.key with device.key
  - replace contents of ca.cert with rootCA.pem

### Installing Certs on a Cisco IOS XE Device

The following example shows how to install certs on a Cisco IOS XE device:

```

# Send:
Device# configure terminal
Device(config)# crypto pki import trustpoint1 pem terminal password password1

```



```

# Receive:
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.

# Send:
# Contents of rootCA.pem, followed by newline + 'quit' + newline:
-----BEGIN CERTIFICATE-----
<snip>
-----END CERTIFICATE-----
quit

# Receive:
% Enter PEM-formatted encrypted private General Purpose key.
% End with "quit" on a line by itself.

# Send:
# Contents of device.des3.key, followed by newline + 'quit' + newline:
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,D954FF9E43F1BA20
<snip>
-----END RSA PRIVATE KEY-----
quit

# Receive:
% Enter PEM-formatted General Purpose certificate.
% End with a blank line or "quit" on a line by itself.

# Send:
# Contents of device.crt, followed by newline + 'quit' + newline:
-----BEGIN CERTIFICATE-----
<snip>
-----END CERTIFICATE-----
quit

# Receive:
% PEM files import succeeded.
Device(config)#

# Send:
Device(config)# crypto pki trustpoint trustpoint1
Device(ca-trustpoint)# revocation-check none
Device(ca-trustpoint)# end
Device#

```

## Enable Secure gNMI communication between Device and Crosswork Data Gateway

### Secure gNMI set up workflow:

1. Upload the trust chain to the Crosswork Certificate Management UI in Cisco Crosswork. See [Configure gNMI Certificate, on page 17](#).
2. Update device configuration with secure gNMI port details from Cisco Crosswork UI. See [Device Configuration from Cisco Crosswork UI, on page 19](#)

### Configure gNMI Certificate

Crosswork Data Gateway acts as the gNMI client while the device acts as gNMI server. Crosswork Data Gateway validates the device using a trust chain. It is expected that you have a global trust chain for all the devices. If you have multiple trust chains, add all the device trust chains (single or multiple vendors) in a

single .pem file and upload this .pem file to the Crosswork Certificate Management UI. For sample device configuration to configure trust point on devices, see [Sample Device Configuration - gNMI, on page 14](#).



**Note** You can upload only one gNMI Certificate to Crosswork.

To configure the gNMI Certificate:

**Step 1** From the Cisco Crosswork UI, go to **Administration > Certificate Management**.

**Step 2** Click the + icon to add certificate.

**Step 3** In **Add Certificate** window, enter the following details:

- **Device Certificate Name** - Enter a name for the certificate.
- **Certificate Role** - Select **Device gNMI Communication**.
- **Device Trust Chain** - Browse your local file system to the location of the .pem file and select the file.

[Home](#) / [Administration](#) / [Certificate Management](#) / [Add Certificate](#)

## Add Certificate

**Certificate Name \***

**Certificate Role \***

**Device Trust Chain \***

Save

Cancel

**Note** If gNMI certificate is already configured and you wish to onboard a device with a different trust chain, update the existing .pem file to include details of the new CA. Select the existing gNMI certificate from the list, click the Edit icon and upload the new .pem file.

**Step 4** Click **Save**.

The gNMI Certificate is listed in the configured certificates once it has been added successfully.

The screenshot shows the Cisco Crosswork Network Automation interface. The left sidebar contains navigation icons for Home, Device Management, and Administration. The main content area is titled 'Certificates' and shows a table of configured certificates. The table has two columns: 'Name' and 'Expiration Date'. The first row in the table is 'Device-gNMI-Certs' with an expiration date of 'Fri, Jan 7, 2022, 3:31:...'.

	Name	Expiration Date
<input type="checkbox"/>	Device-gNMI-Certs	Fri, Jan 7, 2022, 3:31:...
<input type="checkbox"/>	Crosswork-Internal-Communic...	Sun, Jan 22, 2023, 7:..
<input type="checkbox"/>	Crosswork-ZTP-Device-SUDI	Mon, May 14, 2029, 1.
<input type="checkbox"/>	Crosswork-ZTP-Owner	Sun, Jan 22, 2023, 7:..

**Device Configuration from Cisco Crosswork UI**

Once you have configured the gNMI certificate in the Crosswork UI, update the device with secure protocol details.

1. From the Cisco Crosswork UI, navigate to **Device Management** > **Network Devices**
2. Select the device and click **Edit** to update the **Protocol** field with the details as:  
**Protocol** for secure communication : **GNMI\_SECURE Port**.

Edit Device Details
✕

▼ General

<p><b>Configured State*</b> <input type="text" value="DOWN"/></p> <p><b>Reachability Check*</b> <input type="text" value="ENABLE"/></p> <p><b>Credential Profile*</b> <input type="text" value="xrvr"/></p> <p><b>Host Name</b> <input type="text" value="xrvr2"/></p> <p><b>Inventory ID</b> <input type="text"/></p> <p><b>Data Gateway</b> <input type="text" value="None"/></p> <p><b>Software Type</b> <input type="text" value="IOS XR"/></p> <p><b>Software Version</b> <input type="text" value="6.6.2"/></p>	<p><b>UUID</b> <input type="text" value="3166bf90-bbbd-4d19-933e-817caacfa"/></p> <p><b>Serial Number</b> <input type="text"/></p> <p><b>Mac Address</b> <input type="text"/></p> <p><b>Capability*</b> <input type="text" value="SNMP, YANG_CLI"/></p> <p><b>Tags</b> <input type="text"/></p> <p><b>Product Type</b> <input type="text" value="CISCO-XRv9000"/></p> <p><b>Syslog Format</b> <input type="text" value="UNKNOWN"/></p>
---	--

▼ Connectivity Details

Protocol *	IP Address / Subnet Mask *	Port *	Timeout	Encoding Type *	
<input type="text" value="SSH"/>	<input type="text" value="10.11.0.11 / 16"/>	<input type="text" value="22"/>	<input type="text" value="30"/>	<input type="text"/>	<input type="text" value="🗑️"/>
<input type="text" value="SNMP"/>	<input type="text" value="10.11.0.11 / 16"/>	<input type="text" value="161"/>	<input type="text" value="30"/>	<input type="text"/>	<input type="text" value="🗑️"/>
<input type="text" value="GNMI_SECURE"/>	<input type="text" value="10.11.0.11 / 16"/>	<input type="text" value="57400"/>	<input type="text" value="1500"/>	<input type="text" value="PROTO"/>	<input type="text" value="🗑️"/>

[+ Add Another](#)

> Routing Info

## Syslog Collection Job

Cisco Crosswork Data Gateway supports Syslog-based events collection from devices. Following Syslog formats are supported:

- RFC5424 syslog format
- RFC3164 syslog format



**Note** Syslog must be configured on devices before you can submit Syslog collection jobs. Please refer to the platform-specific documentation.

For sample device configuration, see [Configure Syslog in RFC3164/RFC5424 format, on page 21](#).

Following is a sample Syslog collection payload:

```
{
  "collection_job": {
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "c6f25a33-92e6-468a-ba0d-15490f1ce787"
          ]
        }
      }
    }
  }
}
```

```

    },
    "sensor_output_configs": [
      {
        "sensor_data": {
          "syslog_sensor": {
            "pris": {
              "facilities": [0, 1, 3, 23,4],
              "severities": [0, 4, 5, 6, 7]
            }
          }
        },
        "destination": {
          "context_id": "syslogtopic",
          "destination_id": "c2a8fba8-8363-3d22-b0c2-a9e449693fae"
        }
      }
    ],
    "sensor_input_configs": [
      {
        "sensor_data": {
          "syslog_sensor": {
            "pris": {
              "facilities": [0,1, 3, 23,4],
              "severities": [0,4, 5, 6, 7]
            }
          }
        },
        "cadence_in_millisecc": "60000"
      }
    ],
    "application_context": {
      "context_id": "demomilesstone2syslog",
      "application_id": "SyslogDemo2"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "SYSLOG_COLLECTOR"
    }
  }
}

```

Based on the facilities and severities mentioned in the payload, matching Syslog events will be sent to the specified destination. All other non-matching syslog events will be dropped.

## Configure Syslog in RFC3164/RFC5424 format

This section lists sample configuration to configure syslog in the RFC3164 or RFC5424 format on the device. The same configuration can also be used for non-secure syslog configuration on the device.

### Configure RFC3164 Syslog format



**Note** The configuration highlighted in the code below is required to avoid formatting issues in the parsed output.

For Cisco IOS XR devices:

```

logging <server 1> port 9514 OR logging <server 1> vrf <vrfname> port 9514
logging trap [severity]
logging facility [facility value]
logging suppress duplicates

```

```
service timestamps log datetime msec show-timezone year
logging hostnameprefix <some host related prefix e.g.iosxrhost2>
```

For Cisco IOS XE Devices:

```
no logging message-counter syslog
logging trap <serverity>
logging facility <facility>
logging host 172.29.194.174 transport tcp port 9898 session-id string <sessionidstring> -->
  To use TCP channel
OR
logging host 172.29.194.174 transport udp port 9514 session-id string <sessionidstring>
---> To use UDP channel
OR
logging host <cdg ip> vrf Mgmt-intf transport udp port 9514 session-id string
<sessionidstring> --> To use UDP via vrf
service timestamps log datetime msec year show-timezone
```

### Configure RFC5424 Syslog format

For Cisco IOS XR devices:

```
logging <server 1> port 9514 OR logging <server 1> vrf <vrfname> port 9514
logging trap [severity]
logging facility [facility value]
logging suppress duplicates
service timestamps log datetime msec show-timezone year
logging hostnameprefix <some host related prefix e.g.iosxrhost2>
logging format rfc5424
```

For Cisco IOS XE Devices:

```
no logging message-counter syslog
logging trap <serverity>
logging facility <facility>
logging host 172.29.194.174 transport tcp port 9898 session-id string <sessionidstring> -->
  To use TCP channel
OR
logging host 172.29.194.174 transport udp port 9514 session-id string <sessionidstring>
---> To use UDP channel
OR
logging host <cdg ip> vrf Mgmt-intf transport udp port 9514 session-id string
<sessionidstring> --> To use UDP via vrf
service timestamps log datetime msec year show-timezone
logging trap syslog-format 5424 --> if applicable
```

## Configure Secure Syslog on Device

Follow these steps to establish a secure syslog communication to the device.

1. Download the Crosswork trust chain from the Certificate Management UI page in Cisco Crosswork.
2. Configure devices with the Crosswork trust chain for syslog configuration.

### Download Syslog Certificates

1. In the Cisco Crosswork UI, go to **Administration > Certificate Management**
2. Click *i* in the 'device-syslog' row as shown in the image below

Crosswork Network Automation

Administration / Certificate Management

Crosswork Platform Services in evaluation mode: 88 days left. Contact your Account Team... More

Certificates

	Name	Expiration Date	Last Updated By	Last Update Time	Associations
<input type="checkbox"/>	external-destination	Fri, Oct 15, 2021, 12:54:58 PM PDT	admin	Sun, Jan 24, 2021, 05:25:39 P...	External Destination
<input type="checkbox"/>	grpc-ext-dest	Fri, Oct 15, 2021, 12:54:58 PM PDT	admin	Sun, Jan 24, 2021, 05:46:54 P...	External Destination
<input type="checkbox"/>	gnmi-cert	Thu, Jan 20, 2022, 03:41:15 PM PST	admin	Sun, Jan 24, 2021, 09:00:59 P...	Device gNMI Communication
<input type="checkbox"/>	Crosswork-Internal-Communication	Tue, Jan 24, 2023, 10:28:54 AM PST	Crosswork	Sun, Jan 24, 2021, 10:28:54 A...	Crosswork Internal TLS
<input type="checkbox"/>	Crosswork-ZTP-Device-SUDI	Mon, May 14, 2029, 01:25:42 PM PDT	Crosswork	Sun, Jan 24, 2021, 10:29:14 A...	ZTP SUDI
<input type="checkbox"/>	Crosswork-ZTP-Owner	Tue, Jan 24, 2023, 10:29:12 AM PST	Crosswork	Sun, Jan 24, 2021, 10:29:12 A...	Secure ZTP Provisioning
<input type="checkbox"/>	device-syslog	Tue, Jan 24, 2023, 10:29:20 AM PST	Crosswork	Sun, Jan 24, 2021, 10:29:20 A...	Device Syslog Communication
<input type="checkbox"/>	Crosswork-Web-Cert	Fri, Jan 23, 2026, 10:27:54 AM PST	Crosswork	Sun, Jan 24, 2021, 10:27:54 A...	Crosswork Web Server

3. Click **Export All** to download the certificates.

device-syslog Certificate

Description Crosswork Device Root CA

Signed CISCO SYSTEMS INC

Installed Sun Jan 24 18:29:20 UTC 2021

Signed By Crosswork Device Root CA

Expires Fri Jan 23 18:29:18 UTC 2026

---

Description device-syslog

Signed CISCO SYSTEMS INC

Installed Sun Jan 24 18:29:20 UTC 2021

Signed By Crosswork Device Root CA

Expires Tue Jan 24 18:29:20 UTC 2023

---

Description PRIVATE KEY

Signed

Installed Sun Jan 24 18:29:20 UTC 2021

Signed By

Expires

**Export All** Cancel

The following files are downloaded to your system.

Name
intermediate.key
intermediate.crt
ca.crt

## Syslog Configuration on Device

### Sample XR Device Configuration to enable TLS

```
RP/0/RSP0/CPU0:ASR9k(config)#crypto ca trustpoint syslog-root
RP/0/RSP0/CPU0:ASR9k(config-trustp)#enrollment terminal
RP/0/RSP0/CPU0:ASR9k(config-trustp)#crl optional
RP/0/RSP0/CPU0:ASR9k(config-trustp)#commit
RP/0/RSP0/CPU0:ASR9k(config-trustp)#end
RP/0/RSP0/CPU0:ASR9k#
RP/0/RSP0/CPU0:ASR9k#crypto ca authenticate syslog-root
Fri Jan 22 11:07:41.880 GMT
```

Enter the base 64 encoded certificate.  
End with a blank line or the word "quit" on a line by itself

```
-----BEGIN CERTIFICATE-----
MIIGKzCCBBOgAwIBAgIRAKfyU89yjmrxVDRKBWuSGPgWdQYJKoZIhvcNAQELBQAw
bDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAkNBMRERwDwYDVQQHEWhTYW4gSm9zZTEa
.....
jPQ/UrO8N3sC1gGJX7CIIh5cE+KIJ51ep8ileKSJ5wHWRtmv342MnG2StgOTtaFF
vrkWHd02o6jRuYXDWEuptDOg8oEritZb+SNPXWUc/2mbYog6ks6EeMC69VjkZPo=
-----END CERTIFICATE-----
```

Read 1583 bytes as CA certificate

Serial Number : A7:F2:53:CF:72:8E:6A:D7:54:34:4A:05:6B:92:18:F8

Subject:

CN=Crosswork Device Root CA,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US

Issued By :

CN=Crosswork Device Root CA,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US

Validity Start : 02:37:09 UTC Sat Jan 16 2021

Validity End : 02:37:09 UTC Thu Jan 15 2026

SHA1 Fingerprint:

209B3815271C22ADF78CB906F6A32DD9D97BBDBA

Fingerprint: 2FF85849EBAAB9B059ACB9F5363D5C9CDo you accept this certificate? [yes/no]: yes

```
RP/0/RSP0/CPU0:ASR9k#config
RP/0/RSP0/CPU0:ASR9k(config)#crypto ca trustpoint syslog-inter
RP/0/RSP0/CPU0:ASR9k(config-trustp)#enrollment terminal
RP/0/RSP0/CPU0:ASR9k(config-trustp)#crl optional
RP/0/RSP0/CPU0:ASR9k(config-trustp)#commit
RP/0/RSP0/CPU0:ASR9k#crypto ca authenticate syslog-inter
Fri Jan 22 11:10:30.090 GMT
```

Enter the base 64 encoded certificate.  
End with a blank line or the word "quit" on a line by itself

```
-----BEGIN CERTIFICATE-----
MIIGFDCCA/ygAwIBAgIRAKhqHQXcJzQzeQK6U2wn8PIwDQYJKoZIhvcNAQELBQAw
bDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAkNBMRERwDwYDVQQHEWhTYW4gSm9zZTEa
.....
51Bk617z6cxFER5c+/PmJFhcreisTxXglaJbFdnB5C8f+0uUIdLghykQ/zaZGuBn
AAB70c9r9OeKJWzvv1e2U8HH1pdQ/nd
-----END CERTIFICATE-----
```

Read 1560 bytes as CA certificate

Serial Number : 02:48:6A:1D:05:DC:27:34:33:79:02:BA:53:6C:27:F0:F2

Subject:

CN=device-syslog,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US

Issued By :

CN=Crosswork Device Root CA,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US



```

Validity Start : 02:37:11 UTC Sat Jan 16 2021
Validity End   : 02:37:11 UTC Mon Jan 16 2023
SHA1 Fingerprint:
                B06F2BFDE95413A8D08A01EE3511BC3D42F01E59

```

```

CA Certificate validated using issuer certificate.
RP/0/RSP0/CPU0:ASR9k#show crypto ca certificates
Fri Jan 22 15:45:17.196 GMT

```

```
Trustpoint      : syslog-root
=====
```

```
CA certificate
```

```

Serial Number  : A7:F2:53:CF:72:8E:6A:D7:54:34:4A:05:6B:92:18:F8
Subject:
    CN=Crosswork Device Root CA,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US
Issued By      :
    CN=Crosswork Device Root CA,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US
Validity Start : 02:37:09 UTC Sat Jan 16 2021
Validity End   : 02:37:09 UTC Thu Jan 15 2026
SHA1 Fingerprint:
                209B3815271C22ADF78CB906F6A32DD9D97BBDBA

```

```
Trustpoint      : syslog-inter
=====
```

```
CA certificate
```

```

Serial Number  : 02:48:6A:1D:05:DC:27:34:33:79:02:BA:53:6C:27:F0:F2
Subject:
    CN=device-syslog,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US
Issued By      :
    CN=Crosswork Device Root CA,O=CISCO SYSTEMS INC,L=San Jose,ST=CA,C=US
Validity Start : 02:37:11 UTC Sat Jan 16 2021
Validity End   : 02:37:11 UTC Mon Jan 16 2023
SHA1 Fingerprint:
                B06F2BFDE95413A8D08A01EE3511BC3D42F01E59

```

```

RP/0/RSP0/CPU0:ASR9k(config)#logging tls-server syslog-tb131
RP/0/RSP0/CPU0:ASR9k(config-logging-tls-peer)#tls-hostname 10.13.0.159
RP/0/RSP0/CPU0:ASR9k(config-logging-tls-peer)#trustpoint syslog-inter
RP/0/RSP0/CPU0:ASR9k(config-logging-tls-peer)#severity debugging
RP/0/RSP0/CPU0:ASR9k(config-logging-tls-peer)#vrf default
RP/0/RSP0/CPU0:ASR9k(config-logging-tls-peer)#commit
RP/0/RSP0/CPU0:ASR9k(config-logging-tls-peer)#exit
RP/0/RSP0/CPU0:ASR9k(config)#exit
RP/0/RSP0/CPU0:ASR9k#exit
RP/0/RSP0/CPU0:ASR9k#show running-config logging
Fri Jan 22 11:17:19.385 GMT
logging tls-server syslog-tb131
vrf default
severity debugging
trustpoint syslog-inter
tls-hostname <CDG Southbound IP>
!
logging trap debugging
logging format rfc5424
logging facility user
logging hostnameprefix ASR9k
logging suppress duplicates

RP/0/RSP0/CPU0:ASR9k#

```

### Sample XE Device Configuration to enable TLS

```

csr8kv(config)#crypto pki trustpoint syslog-root
csr8kv(ca-trustpoint)#enrollment terminal

```

```

csr8kv(ca-trustpoint)#revocation-check none
csr8kv(ca-trustpoint)#chain-validation stop
csr8kv(ca-trustpoint)#end
csr8kv(config)#crypto pki authenticate syslog-root

```

Enter the base 64 encoded CA certificate.  
End with a blank line or the word "quit" on a line by itself

```

-----BEGIN CERTIFICATE-----
MIIFPjCCAyYCCQCO6pK5AOGYdjANBgkqhkiG9w0BAQsFADBhMQswCQYDVQQGEwJV
UzELMAkGA1UECAwCQ0EwETAPBgNVBACMCElpbHBpdGFzMQ4wDAYDVQQKDAVDaXNj
.....
JbimOpXAncoBLo14DXOJLvMVRjn1EULE9AXXCnfnrnBx7jL4CV+qHgEtF6oqclFW
JEA=
-----END CERTIFICATE-----

```

Certificate has the following attributes:

```

    Fingerprint MD5: D88D6D8F E53750D4 B36EB498 0A435DA1
    Fingerprint SHA1: 649DE822 1C222C1F 5101BEB8 B29CDF12 5CEE463B

```

```

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported

```

```

csr8kv(config)#crypto pki trustpoint syslog-intermediate
csr8kv(ca-trustpoint)#enrollment terminal
csr8kv(ca-trustpoint)#revocation-check none
csr8kv(ca-trustpoint)#chain-validation continue syslog-root
csr8kv(ca-trustpoint)#end
csr8kv(config)#crypto pki authenticate syslog-intermediate

```

Enter the base 64 encoded CA certificate.  
End with a blank line or the word "quit" on a line by itself

```

-----BEGIN CERTIFICATE-----
MIIFfTCCA2WgAwIBAgICEAAwDQYJKoZIhvcNAQELBQAwXDELMAkGA1UEBhMCVVMx
EzARBgNVBAGMCKNhbgGmb3JuaWEwExDjAMBgNVBAoMBUNpc2NvMQ4wDAYDVQQQLDAV
.....
Nmz6NQynD7bxdQa9Xq9kyPuY3ZVKXkf312IRH0MEy2yFX/tAen9JqOeZlg8canmw
TxsWA5TLzylRmxqQh88f0CM=
-----END CERTIFICATE-----

```

Trustpoint 'syslog-intermediate' is a subordinate CA.  
but certificate is not a CA certificate.

Manual verification required

Certificate has the following attributes:

```

    Fingerprint MD5: FE27BDBE 9265208A 681670AC F59A2BF1
    Fingerprint SHA1: 03F513BD 4BEB689F A4F4E001 57EC210E 88C7BD19

```

```

csr8kv(config)#logging host <CDG Southbound IP> transport tls port 6514
csr8kv(config)#logging trap informational syslog-format rfc5424
csr8kv(config)#logging facility user
csr8kv(config)#service timestamps log datetime msec year show-timezone

```

```

csr8kv(config)#logging tls-profile tlsv12

```

## Syslog Collection Job Output

When you add a device from Cisco Crosswork UI (**Device Management > Network Devices > Device Details**), the value you choose in the **Syslog Format** field configures the format in which syslog events

received from the device should be parsed by the Syslog Collector. You can choose either **UNKNOWN**, **RFC5424** or **RFC3164**.

Following is the sample output for each of the options:

### 1. UNKNOWN - Syslog Collection Job output contains the syslog event as received from device.



**Note** If the device is configured to generate syslog events in RFC5424/RFC3164 format but no format is specified in the **Syslog Format** field, by default this is considered as **UNKNOWN**.

Sample output:

```
node_id_str: "xrv9k-VM8"
node_id_uuid: ":i\300\216>\366BM\262\270@\337\225\2723&"
collection_id: 1056
collection_start_time: 1616711596200
msg_timestamp: 1616711596201
data_gpbkv {
  timestamp: 1616711596201
  name: "syslogsensor.path"
  fields {
    name: "RAW"
    string_value: "<6>1 Mar 25 15:34:41.321 PDT - SSHD_69570 - - 98949:
RP/0/RP0/CPU0:SSHD_[69570]: %SECURITY-SSHD-6-INFO_SUCCESS : Successfully authenticated
user \'admin\' from \'40.40.40.116\' on \'vty0\' (cipher \'aes128-ctr\', mac \'hmac-sha1\')
\n"
  }
  fields {
    name: "DEVICE_IP"
    string_value: "40.40.40.30"
  }
}
collection_end_time: 1616711596200
collector_uuid: "17328736-b726-4fe3-b922-231a4a30a54f:SYSLOG_COLLECTOR"
status {
  status: SUCCESS
}
model_data {
}
sensor_data {
  syslog_sensor {
    pris {
      facilities: 0
      facilities: 3
      facilities: 4
      facilities: 23
      severities: 0
      severities: 5
      severities: 6
      severities: 7
    }
  }
}
application_contexts {
  application_id: "SyslogApp-xr-8-job1"
  context_id: "xr-8-job1"
}
version: "1"
```

- RFC5424** - If the device is configured to generate syslog events in RFC5424 format and the RFC5424 format is selected in the **Syslog Format** field, the Syslog Job Collection output contains syslog events as received from device ((raw) and the RFC5424 best-effort parsed syslog event from the device.



**Note** The syslog collector will parse the syslog event(best effort parsing) as per the following Java RegEx pattern:

## RFC5424

```
"^<(?!<pri>\\d+)>(?!<version>\\d{1,3})\\s*(?!<date>([0-9]{4}\\s+
9T:.-Z-]+))\\s*(?!<host>\\S+)\\s*(?!<processname>\\S+)\\s*(?!<pr
<message>.+)$";
```

Sample output:

....  
....

```
collection_start_time: 1596307542398
msg_timestamp: 1596307542405
data_gpbkv {
  timestamp: 1596307542405
  name: "syslogsensor.path"
  fields {
    name: "RAW"
    string_value: "<13>1 2020 Aug 1 12:03:32.461 UTC: iosxr254node config 65910 - -
2782: RP/0/RSP0/CPU0:2020 Aug 1 12:03:32.461 UTC: config[65910]: %MGBL-SYS-5-CONFIG_I
: Configured from console by admin on vty0 (10.24.88.215) \n"
  }
  fields {
    name: "RFC5424"
    string_value: "pri=13, severity=5, facility=1, version=1,
date=2020-08-01T12:03:32.461, remoteAddress=/172.28.122.254, host='iosxr254node',
message='2782: RP/0/RSP0/CPU0:2020 Aug 1 12:03:32.461 UTC: config[65910]:
%MGBL-SYS-5-CONFIG_I : Configured from console by admin on vty0 (10.24.88.215) \',
messageId=null, processName=config, structuredDataList=null"
  }
  fields {
    name: "DEVICE_IP"
    string_value: "172.28.122.254"
  }
}
collection_end_time: 1596307542404
collector_uuid: "ac961b09-8f67-4c93-a99a-31eef50f7fa9:SYSLOG_COLLECTOR"
status {
  status: SUCCESS
}
...
...
```



If the Syslog Collector is unable to parse the syslog events according to the format specified in the **Syslog Format** field, then the Syslog Collection Job output contains the syslog events as received from device.

## Create a Collection Job

Follow the steps to create a collection job:




**Note** Collection jobs created through the Cisco Crosswork UI page can only be published once.

### Before you begin

Ensure that a data destination is created (and active) to deposit the collected data. Also, have details of the sensor path and MIB that you plan to collect data from.

**Step 1** From the main menu, go to **Administration > Collection Jobs**.

**Step 2** In the **Collection Jobs** pane on the left hand side, click  button.

**Step 3** In the **Job details** page, enter values for the following fields:

CDG Jobs / New Collection Job

Job Details    Select Devices    Sensor Details    Confirm

Job Details

Application ID\*

Context ID\*

Collection Mode

Collector Type\*

Cancel    Next

- Application ID: A unique identifier for the application.
- Context: A unique identifier to identify your application subscription across all collection jobs.
- Collector Type: Select the type of collection - CLI or SNMP.

Click **Next**.

**Step 4** Select the devices from which the data is to be collected. You can either select based on device tag or manually. Click **Next**.

CDG Jobs / New Collection Job

Job Details    Select Devices    Sensor Details    Confirm

Select By  Select Device Tag  Select Device Manually

Tags will be resolved dynamically at runtime to determine constituent devices.

Select Tags\*

Default

- Mdt(0)
- Smp(2)
- Cl(2)
- Epm(0)
- Reach-Check(2)
- See More

Polling

- Te-Tunnel-tel(1)

Tag Selected

Select tags from the left panel to preview the devices

Cancel    Previous    Next

**Step 5** (Applicable only for CLI collection) Enter the following sensor details:

- Select data destination from **Select Data Destination** drop-down.
- Select sensor type from **Sensor Types** pane on the left.

If you selected **CLI PATH**, Click **+** button and enter the following parameters in the **Add CLI Path** dialog box:

- Collection Cadence: Push or poll cadence in seconds.
- Command: CLI command
- Topic: Topic associated with the output destination.

If you selected **Device Package**, click **+** button and enter values for the following parameters in the **Add Device Package Sensor** dialog box:



- Collection cadence: Push or poll cadence in seconds.
- Device Package Name: Custom XDE device package ID used while creating device package.
- Function name: Function name within custom XDE device package.
- Topic: Topic associated with the output destination.

Enter Key and String value for the paramters.

Click **Save**.


**Step 6** (Applicable only for SNMP collection) Enter the following sensor details:

- Select data destination from **Select Data Destination** drop-down.
- Select sensor type from **Sensor Types** pane on the left.

If you selected **SNMP MIB**, Click **+** button and enter the following parameters in the **Add SNMP MIB** dialog box:

The screenshot shows the 'Add SNMP MIB' dialog box. The 'Collection Cadence' field is set to 60, with a note 'In seconds'. The 'OID' field is empty. The 'Operation' field is a dropdown menu. The 'Topic' field is empty. The background shows the 'Sensor Details' section with 'SNMP MIB' selected under 'Sensor Types'.

- Collection Cadence: Push or poll cadence in seconds.
- OID
- Operation: Select the operation from the list.
- Topic: Topic associated with the output destination.

If you selected **Device Package**, click  button and enter values for the following parameters in the **Add Device Package Sensor** dialog box:

The screenshot shows the 'Add Device Package Sensor' dialog box. The 'Collection Cadence' field is set to 60, with a note 'In seconds'. The 'Device Package Name' field is a dropdown menu. The 'Function Name' field is empty. The 'Topic' field is empty. There is an 'Add Parameters' section with 'Key' and 'String Value' fields. The background shows the 'Sensor Details' section with 'Device Package' selected under 'Sensor Types'.

- Collection Cadence: Push or poll cadence in seconds.
- Device Package Name: Custom device package ID used while creating device package.
- Function name: Function name within custom device package.
- Topic: Topic associated with the output destination.

Enter Key and String value for the paramters.

Click **Save**.

**Step 7** Click **Create Collection Job**.

**Note** When a collection job is submitted for an external Kafka destination i.e., unsecure Kafka, the dispatch job to Kafka fails to connect. The error seen in collector logs is  
`org.apache.kafka.common.errors.TimeoutException: Topic cli-job-kafka-unsecure not present in metadata after 60000 ms. In Kafka logs, the error seen is SSL authentication error "[2021-01-08 22:17:03,049] INFO [SocketServer brokerId=0] Failed authentication with /80.80.80.108 (SSL handshake failed) (org.apache.kafka.common.network.Selector).`

This happens because port is blocked on external Kafka VM. You can use the following command to check if port is listening on Kafka docker/server port:

```
netstat -tulpn
```

Reboot the Kafka VM to fix this issue.

---

## Delete a Collection Job

System and Crosswork Change Automation and Health Insights collection jobs should not be deleted as it will cause issues. Only external collection jobs can be deleted from the **Collection Jobs** page.

Follow the steps to delete a collection job:

---

**Step 1** Go to **Administration > Collection Jobs**.

**Step 2** In the **Collection Jobs** pane on the left hand side, select the collection job you want to delete.

**Step 3** Click delete button.

**Step 4** Click **Delete** button when prompted.

---

## Monitoring Collection Jobs

You can monitor the status of the collection jobs currently active on all the Cisco Crosswork Data Gateway instances enrolled with Cisco Crosswork from the **Collection Jobs** page.

From the Cisco Crosswork main menu, choose **Administration > Collection Jobs**.

The screenshot shows the 'Collection Jobs' monitoring interface. The left pane displays a list of collection jobs with columns for Status, App ID, and Context ID. The right pane shows the details for a specific job, including its status (Successful), last modified time, and a flow diagram illustrating the data path from Devices through Collections, Data Gateways, Distributions, and Destinations. Below the flow diagram, there is a table for 'Collection Issues' with columns for Collection Status, Hostname, Device Id, Sensor Data, and Last Reported Time. This table currently shows 'No Rows To Show'.

The **Collection Jobs** pane shows the list of all active collection jobs along with their Status, App ID, and Context ID.


The **Job Details** pane shows the details of a particular job selected in the **Collection Jobs** pane.

When you select a job, more details are displayed in the **Job Details** pane:




- Application name and context associated with the collection job.
- Status of the collection job.

**Note**

- Once a device is mapped to a Cisco Crosswork Data Gateway, the status of all the associated collection jobs is set to 'Unknown'. A job could have status as 'Unknown' for either of the following reasons:
  - Cisco Crosswork Data Gateway has not yet reported its status.
  - Loss of connection between Cisco Crosswork Data Gateway and Crosswork.
  - Cisco Crosswork Data Gateway received the collection job, but actual collection is still pending.
- After the collection job is processed, the status changes to 'Successful' if the processing was successful or else it changes to 'Failed'.
- If a collection job is in degraded state, one of the reasons might be that the static routes to the device have been erased from Crosswork Data Gateway.
- Health Insights - KPI jobs must be enabled only on devices mapped to an extended Crosswork Data Gateway VM. Health Insights - KPI jobs that are enabled on devices mapped to standard Crosswork Data Gateway VM will have the job status as Degraded and the collection status as Failed.

- Job configuration of the collection job that you pass in the REST API request. Click  icon next to **Config Details** to view the job configuration. Cisco Crosswork lets you view configuration in two modes:
  - View Mode
  - Text Mode
- Collection type
- Time and date of last modification of the collection job.
- Collections (x): x refers to requested input collections that span device by sensor paths. The corresponding (y) **Issues** is the count of input collections that are in UNKNOWN or FAILED state.
- Distributions (x): x refers to requested output collections that span device by sensor paths. The corresponding (y) **Issues** is the count of output collections that are in UNKNOWN or FAILED state.


Cisco Crosswork also displays the following details for collections and distributions:

Field	Description
Collection/Distribution Status	Status of the collection/distribution. It is reported on a on change basis from Crosswork Data Gateway.  Click  next to the collection/distribution status for details.
Hostname	Device hostname with which the collection job is associated.
Device Id	Unique identifier of the device from which data is being collected.
Sensor Data	<p>Sensor path</p> <p>Click  to see collection/distribution summary. From the sensor data summary pop up you can copy the sensor data by clicking <b>Copy to Clipboard</b>.</p> <p>Click  to see collection/distribution metrics summary. The metrics are reported on cadence-basis i.e., once every 10 minutes by default. It shows the following metrics for a collection:</p> <ul style="list-style-type: none"> <li>• last_collection_time_msec</li> <li>• total_collection_message_count</li> <li>• last_device_latency_msec</li> <li>• last_collection_cadence_msec</li> </ul> <p>It shows the following metrics for a collection:</p> <ul style="list-style-type: none"> <li>• total_output_message_count</li> <li>• last_destination_latency_msec</li> <li>• last_output_cadence_msec</li> <li>• last_output_time_msec</li> <li>• total_output_bytes_count</li> </ul>
Destination	Data destination for the job.
Last Status Change Reported Time	Time and date on which last status change was reported for that device sensor pair from Crosswork Data Gateway

**Note**

- `Create Failed` error means out of N devices, some devices failed to setup. However, the collection would happen on the devices that were successfully setup. You can identify the device(s) causing this error by using `Control Status` API.
- If job creation failed on a particular device because of NSO errors, after fixing NSO errors, you have to manually change the administration state of the device first to "Down" and then "Up". However, doing so resets the collection on the device.

**Note**

Create/Delete failed errors are shown in a different screen pop up. Click  next to the job status to see details of the error.

- You may also try recreating the job using PUT collection job API with the same payload.

## List of Pre-loaded Traps and MIBs for SNMP Collection

This section lists the traps and MIBs that the Cisco Crosswork Data Gateway supports for SNMP collection.

**Note**

This list is applicable only when Crosswork is the target application and is not limited when the target is an external application.

Note the following constraints:

- The system cannot extract index values from OIDs of conceptual tables. If any of the columns that define indices in the conceptual table are not populated, the index value is replaced on the data plane with the instance identifier (oid suffix) of the row.
- The system cannot extract index values from conceptual tables that include the **AUGMENT** keyword or refer to indices of other tables.
- Named-number enumerations (using the integer syntax) are sent on the wire using their numeric value.

**Table 2: Supported Traps**

Trap	OID
linkDown	1.3.6.1.6.3.1.1.5.3
linkUp	1.3.6.1.6.3.1.1.5.4
coldStart	1.3.6.1.6.3.1.1.5.1
isisAdjacencyChange	1.3.6.1.2.1.138.0.17

ADSL-LINE-MIB.mib	CISCO-LWAPP- INTERFACE-MIB.mib	IANA-ITU-ALARM- TC-MIB.mib
ADSL-TC-MIB.mib	CISCO-LWAPP- IPS-MIB.mib	IANA-LANGUAGE- MIB.mib
AGENTX-MIB.mib	CISCO-LWAPP- LINKTEST-MIB.mib	IANA-RTPROTO- MIB.mib
ALARM-MIB.mib	CISCO-LWAPP- LOCAL-AUTH-MIB.mib	IANAifType-MIB.mib
APS-MIB.mib	CISCO-LWAPP- MDNS-MIB.mib	IEEE8021-CFM-MIB.mib
ATM-FORUM-MIB.mib	CISCO-LWAPP- MESH-BATTERY-MIB.mib	IEEE8021-PAE-MIB.mib
ATM-FORUM- TC-MIB.mib	CISCO-LWAPP- MESH-LINKTEST-MIB.mib	IEEE8021-TC-MIB.mib
ATM-MIB.mib	CISCO-LWAPP- MOBILITY-EXT-MIB.mib	IEEE802171-CFM- MIB.mib
ATM-TC-MIB.mib	CISCO-LWAPP- MOBILITY-MIB.mib	IEEE8023-LAG-MIB.mib
ATM2-MIB.mib	CISCO-LWAPP- NETFLOW-MIB.mib	IEEE802dot11-MIB.mib
BGP4-MIB.mib	CISCO-LWAPP- REAP-MIB.mib	IF-INVERTED- STACK-MIB.mib
BRIDGE-MIB.mib	CISCO-LWAPP- RF-MIB.mib	IF-MIB.mib
CISCO-AAA- SERVER-MIB.mib	CISCO-LWAPP- SI-MIB.mib	IGMP-STD-MIB.mib
CISCO-AAA- SESSION-MIB.mib	CISCO-LWAPP- TC-MIB.mib	INET-ADDRESS-MIB.mib
CISCO-AAL5-MIB.mib	CISCO-LWAPP- TRUSTSEC-MIB.mib	INT-SERV-MIB.mib
CISCO-ACCESS- ENVMON-MIB.mib	CISCO-LWAPP- TSM-MIB.mib	INTEGRATED-SERVICES -MIB.mib
CISCO-ATM-EXT -MIB.mib	CISCO-LWAPP- WLAN-MIB.mib	IP-FORWARD-MIB.mib
CISCO-ATM- PVCTRAP-EXTN-MIB.mib	CISCO-LWAPP-WLAN -SECURITY-MIB.mib	IP-MIB.mib
CISCO-ATM- QOS-MIB.mib	CISCO-MEDIA- GATEWAY-MIB.mib	IPMCAST-MIB.mib
CISCO-AUTH- FRAMEWORK-MIB.mib	CISCO-MOTION-MIB.mib	IPMROUTE-MIB.mib
CISCO-BGP-POLICY -ACCOUNTING-MIB.mib	CISCO-MPLS-LSR -EXT-STD-MIB.mib	IPMROUTE-STD -MIB.mib
CISCO-BGP4-MIB.mib	CISCO-MPLS-TC -EXT-STD-MIB.mib	IPV6-FLOW-LABEL -MIB.mib



CISCO-BULK-FILE -MIB.mib	CISCO-MPLS-TE-STD -EXT-MIB.mib	IPV6-ICMP-MIB.mib
CISCO-CBP-TARGET -MIB.mib	CISCO-NAC-TC -MIB.mib	IPV6-MIB.mib
CISCO-CBP-TARGET -TC-MIB.mib	CISCO-NBAR-PROTOCOL -DISCOVERY-MIB.mib	IPV6-MLD-MIB.mib
CISCO-CBP-TC-MIB.mib	CISCO-NETSYNC -MIB.mib	IPV6-TC.mib
CISCO-CCME-MIB.mib	CISCO-NTP-MIB.mib	IPV6-TCP-MIB.mib
CISCO-CDP-MIB.mib	CISCO-OSPF- MIB.mib	IPV6-UDP-MIB.mib
CISCO-CEF-MIB.mib	CISCO-OSPF- TRAP-MIB.mib	ISDN-MIB.mib
CISCO-CEF-TC.mib	CISCO-OTN-IF-MIB.mib	ISIS-MIB.mib
CISCO-CLASS-BASED -QOS-MIB.mib	CISCO-PAE-MIB.mib	ITU-ALARM-MIB.mib
CISCO-CONFIG- COPY-MIB.mib	CISCO-PAGP-MIB.mib	ITU-ALARM-TC- MIB.mib
CISCO-CONFIG- MAN-MIB.mib	CISCO-PIM-MIB.mib	L2TP-MIB.mib
CISCO-CONTENT- ENGINE-MIB.mib	CISCO-PING-MIB.mib	LANGTAG-TC-MIB.mib
CISCO-CONTEXT- MAPPING-MIB.mib	CISCO-POLICY-GROUP -MIB.mib	LLDP-EXT-DOT1 -MIB.mib
CISCO-DATA -COLLECTION-MIB.mib	CISCO-POWER- ETHERNET-EXT-MIB.mib	LLDP-EXT-DOT3 -MIB.mib
CISCO-DEVICE-EXCEPTION -REPORTING-MIB.mib	CISCO-PRIVATE -VLAN-MIB.mib	LLDP-MIB.mib
CISCO-DIAL- CONTROL-MIB.mib	CISCO-PROCESS-MIB.mib	MAU-MIB.mib
CISCO-DOT11- ASSOCIATION-MIB.mib	CISCO-PRODUCTS- MIB.mib	MGMD-STD-MIB.mib
CISCO-DOT11-HT- PHY-MIB.mib	CISCO-PTP-MIB.mib	MPLS-FTN-STD- MIB.mib
CISCO-DOT11-IF-MIB.mib	CISCO-RADIUS- EXT-MIB.mib	MPLS-L3VPN-STD- MIB.mib
CISCO-DOT11-SSID- SECURITY-MIB.mib	CISCO-RF-MIB.mib	MPLS-LDP-ATM- STD-MIB.mib
CISCO-DOT3- OAM-MIB.mib	CISCO-RF-SUPPLEMENTAL -MIB.mib	MPLS-LDP-FRAME -RELAY-STD-MIB.mib
CISCO-DS3-MIB.mib	CISCO-RTTMON-TC -MIB.mib	MPLS-LDP-GENERIC- STD-MIB.mib
CISCO-DYNAMIC- TEMPLATE-MIB.mib	CISCO-SELECTIVE- VRF-DOWNLOAD-MIB.mib	MPLS-LDP-MIB.mib
CISCO-DYNAMIC -TEMPLATE-TC-MIB.mib	CISCO-SESS-BORDER-CTRLR -CALL-STATS-MIB.mib	MPLS-LDP-STD-MIB.mib

CISCO-EIGRP-MIB.mib	CISCO-SESS-BORDER-CTRLR-EVENT-MIB.mib	MPLS-LSR-MIB.mib
CISCO-EMBEDDED-EVENT-MGR-MIB.mib	CISCO-SESS-BORDER-CTRLR-STATS-MIB.mib	MPLS-LSR-STD-MIB.mib
CISCO-ENHANCED-IMAGE-MIB.mib	CISCO-SMI.mib	MPLS-TC-MIB.mib
CISCO-ENHANCED-MEMPOOL-MIB.mib	CISCO-SONET-MIB.mib	MPLS-TC-STD-MIB.mib
CISCO-ENTITY-ASSET -MIB.mib	CISCO-ST-TC.mib	MPLS-TE-MIB.mib
CISCO-ENTITY-EXT -MIB.mib	CISCO-STACKWISE- MIB.mib	MPLS-TE-STD-MIB.mib
CISCO-ENTITY-FRU-CONTROL-MIB.mib	CISCO-STP-EXTENSIONS-MIB.mib	MPLS-VPN-MIB.mib
CISCO-ENTITY- QFP-MIB.mib	CISCO-SUBSCRIBER-IDENTITY-TC-MIB.mib	MSDP-MIB.mib
CISCO-ENTITY-REDUNDANCY-MIB.mib	CISCO-SUBSCRIBER-SESSION-MIB.mib	NET-SNMP-AGENT-MIB.mib
CISCO-ENTITY-REDUNDANCY-TC-MIB.mib	CISCO-SUBSCRIBER-SESSION-TC-MIB.mib	NET-SNMP-EXAMPLES-MIB.mib
CISCO-ENTITY- SENSOR-MIB.mib	CISCO-SYSLOG-MIB.mib	NET-SNMP-MIB.mib
CISCO-ENTITY-VENDORTYPE-OID-MIB.mib	CISCO-SYSTEM-EXT- MIB.mib	NET-SNMP-TC.mib
CISCO-ENVMON-MIB.mib	CISCO-SYSTEM-MIB.mib	NHRP-MIB.mib
CISCO-EPM-NOTIFICATION-MIB.mib	CISCO-TAP2-MIB.mib	NOTIFICATION-LOG-MIB.mib
CISCO-ETHER-CFM- MIB.mib	CISCO-TC.mib	OLD-CISCO-CHASSIS-MIB.mib
CISCO-ETHERLIKE- EXT-MIB.mib	CISCO-TCP-MIB.mib	OLD-CISCO-INTERFACES-MIB.mib
CISCO-FABRIC- C12K-MIB.mib	CISCO-TEMP-LWAPP-DHCP-MIB.mib	OLD-CISCO-SYS- MIB.mib
CISCO-FIREWALL -TC.mib	CISCO-TRUSTSEC -SXP-MIB.mib	OLD-CISCO-SYSTEM-MIB.mib
CISCO-FLASH-MIB.mib	CISCO-TRUSTSEC -TC-MIB.mib	OPT-IF-MIB.mib
CISCO-FRAME- RELAY-MIB.mib	CISCO-UBE-MIB.mib	OSPF-MIB.mib
CISCO-FTP-CLIENT -MIB.mib	CISCO-UNIFIED-COMPUTING-ADAPTOR -MIB.mib	OSPF-TRAP-MIB.mib
CISCO-HSRP-EXT -MIB.mib	CISCO-UNIFIED-COMPUTING-COMPUTE-MIB.mib	OSPFV3-MIB.mib

CISCO-HSRP-MIB.mib	CISCO-UNIFIED-COMPUTING-ETHER -MIB.mib	P-BRIDGE-MIB.mib
CISCO-IETF-ATM2 -PVCTRAP-MIB.mib	CISCO-UNIFIED-COMPUTING-FC- MIB.mib	PIM-MIB.mib
CISCO-IETF-BFD -MIB.mib	CISCO-UNIFIED-COMPUTING-MEMORY -MIB.mib	PIM-STD-MIB.mib
CISCO-IETF-FRR -MIB.mib	CISCO-UNIFIED- COMPUTING -MIB.mib	POWER-ETHERNET -MIB.mib
CISCO-IETF-IPMROUTE -MIB.mib	CISCO-UNIFIED-COMPUTING-NETWORK -MIB.mib	PPP-IP-NCP-MIB.mib
CISCO-IETF-ISIS -MIB.mib	CISCO-UNIFIED-COMPUTING-PROCESSOR -MIB.mib	PPP-LCP-MIB.mib
CISCO-IETF-MPLS-ID -STD-03-MIB.mib	CISCO-UNIFIED-COMPUTING-TC- MIB.mib	PPVPN-TC-MIB.mib
CISCO-IETF-MPLS-TE-EXT-STD-03- MIB.mib	CISCO-VLAN-IFTABLE-RELATIONSHIP -MIB.mib	PTOPO-MIB.mib
CISCO-IETF-MPLS-TE-P2MP-STD-MIB.mib	CISCO-VLAN-MEMBERSHIP-MIB.mib	PerfHist-TC-MIB.mib
CISCO-IETF-MSDP -MIB.mib	CISCO-VOICE-COMMON -DIAL-CONTROL-MIB.mib	Q-BRIDGE-MIB.mib
CISCO-IETF-PIM-EXT -MIB.mib	CISCO-VOICE-DIAL -CONTROL-MIB.mib	RADIUS-ACC-CLIENT -MIB.mib
CISCO-IETF-PIM -MIB.mib	CISCO-VOICE-DNIS -MIB.mib	RADIUS-AUTH-CLIENT -MIB.mib
CISCO-IETF-PW- ATM-MIB.mib	CISCO-VPDN-MGMT -MIB.mib	RFC-1212.mib
CISCO-IETF-PW- ENET-MIB.mib	CISCO-VTP-MIB.mib	RFC-1215.mib
CISCO-IETF-PW-MIB.mib	CISCO-WIRELESS-NOTIFICATION-MIB.mib	RFC1155-SMI.mib
CISCO-IETF-PW- MPLS-MIB.mib	CISCOSB-DEVICEPARAMS -MIB.mib	RFC1213-MIB.mib
CISCO-IETF-PW -TC-MIB.mib	CISCOSB- HWENVIROMENT.mib	RFC1315-MIB.mib
CISCO-IETF-PW -TDM-MIB.mib	CISCOSB-MIB.mib	RFC1398-MIB.mib
CISCO-IETF-VPLS -BGP-EXT-MIB.mib	CISCOSB-Physicaldescription -MIB.mib	RIPv2-MIB.mib
CISCO-IETF-VPLS -GENERIC-MIB.mib	DIAL-CONTROL-MIB.mib	RMON-MIB.mib
CISCO-IETF-VPLS- LDP-MIB.mib	DIFFSERV-DSCP-TC.mib	RMON2-MIB.mib

## List of Pre-loaded Traps and MIBs for SNMP Collection

CISCO-IF-EXTENSION -MIB.mib	DIFFSERV-MIB.mib	RSTP-MIB.mib
CISCO-IGMP-FILTER -MIB.mib	DISMAN-NSLOOKUP -MIB.mib	RSVP-MIB.mib
CISCO-IMAGE-LICENSE -MGMT-MIB.mib	DISMAN-PING-MIB.mib	SMON-MIB.mib
CISCO-IMAGE-MIB.mib	DISMAN-SCHEDULE -MIB.mib	SNA-SDLC-MIB.mib
CISCO-IMAGE-TC.mib	DISMAN-SCRIPT-MIB.mib	SNMP-COMMUNITY -MIB.mib
CISCO-IP-LOCAL- POOL-MIB.mib	DISMAN-TRACEROUTE -MIB.mib	SNMP-FRAMEWORK -MIB.mib
CISCO-IP-TAP-MIB.mib	DOT3-OAM-MIB.mib	SNMP-MPD-MIB.mib
CISCO-IP-URPF-MIB.mib	DRAFT-MSDP-MIB.mib	SNMP-NOTIFICATION -MIB.mib
CISCO-IPMROUTE- MIB.mib	DS0-MIB.mib	SNMP-PROXY-MIB.mib
CISCO-IPSEC-FLOW -MONITOR-MIB.mib	DS1-MIB.mib	SNMP-REPEATER -MIB.mib
CISCO-IPSEC-MIB.mib	DS3-MIB.mib	SNMP-TARGET-MIB.mib
CISCO-IPSEC-POLICY -MAP-MIB.mib	ENTITY-MIB.mib	SNMP-USER-BASED -SM-MIB.mib
CISCO-IPSLA- AUTOMEASURE-MIB.mib	ENTITY-SENSOR-MIB.mib	SNMP-USM-AES -MIB.mib
CISCO-IPSLA- ECHO-MIB.mib	ENTITY-STATE-MIB.mib	SNMP-USM-DH- OBJECTS-MIB.mib
CISCO-IPSLA- JITTER-MIB.mib	ENTITY-STATE- TC-MIB.mib	SNMP-VIEW- BASED-ACM-MIB.mib
CISCO-IPSLA- TC-MIB.mib	ESO-CONSORTIUM -MIB.mib	SNMPv2-CONF.mib
CISCO-ISDN-MIB.mib	ETHER-WIS.mib	SNMPv2-MIB.mib
CISCO-LICENSE- MGMT-MIB.mib	EtherLike-MIB.mib	SNMPv2-SMI.mib
CISCO-LOCAL- AUTH-USER-MIB.mib	FDDI-SMT73-MIB.mib	SNMPv2-TC-v1.mib
CISCO-LWAPP- AAA-MIB.mib	FR-MFR-MIB.mib	SNMPv2-TC.mib
CISCO-LWAPP- AP-MIB.mib	FRAME-RELAY -DTE-MIB.mib	SNMPv2-TM.mib
CISCO-LWAPP- CCX-RM-MIB.mib	FRNETSERV- MIB.mib	SONET-MIB.mib
CISCO-LWAPP- CDP-MIB.mib	GMPLS-LSR- STD-MIB.mib	SYSAPPL-MIB.mib
CISCO-LWAPP-CLIENT -ROAMING-CAPABILITY.mib	GMPLS-TC-STD- MIB.mib	TCP-MIB.mib
CISCO-LWAPP-CLIENT -ROAMING-MIB.mib	GMPLS-TE-STD-MIB.mib	TOKEN-RING-RMON -MIB.mib

CISCO-LWAPP-DHCP -MIB.mib	HC-PerfHist-TC-MIB.mib	TOKENRING-MIB.mib
CISCO-LWAPP-DOT11-CLIENT-CALIB-MIB.mib	HC-RMON-MIB.mib	TRANSPORT-ADDRESS-MIB.mib
CISCO-LWAPP-DOT11-CLIENT-CCX-TC-MIB.mib	HCNUM-TC.mib	TUNNEL-MIB.mib
CISCO-LWAPP-DOT11-LDAP-MIB.mib	HOST-RESOURCES -MIB.mib	UDP-MIB.mib
CISCO-LWAPP- DOT11-MIB.mib	HOST-RESOURCES -TYPES.mib	VPN-TC-STD-MIB.mib
CISCO-LWAPP-DOWNLOAD-MIB.mib	IANA-ADDRESS-FAMILY-NUMBERS-MIB.mib	VRRP-MIB.mib
CISCO-LWAPP-IDS-MIB.mib	IANA-GMPLS-TC-MIB.mib	

## List of Pre-loaded YANG Modules for MDT Collection

This section lists the YANG modules that the Cisco Crosswork Data Gateway supports for MDT collection on Cisco IOS XR devices.

cli_xr_bgp_oper.yang	Cisco-IOS-XR-ip-bfd-oper.yang
Cisco-IOS-XR-ipv4-bgp-oper.yang	Cisco-IOS-XR-asr9k-xbar-oper.yang
Cisco-IOS-XR-ipv4-acl-oper.yang	Cisco-IOS-XR-snmp-sensormib-oper.yang
Cisco-IOS-XR-shellutil-filesystem-oper.yang	Cisco-IOS-XR-config-cfgmgr-oper.yang
Cisco-IOS-XR-infra-alarm-logger-oper.yang	Cisco-IOS-XR-infra-fti-oper.yang
Cisco-IOS-XR-icpe-infra-oper.yang	Cisco-IOS-XR-dot1x-oper.yang
Cisco-IOS-XR-fretta-bcm-dpa-stats-oper.yang	Cisco-IOS-XR-sdr-invmgr-diag-oper.yang
Cisco-IOS-XR-cofo-infra-oper.yang	Cisco-IOS-XR-wanphy-ui-oper.yang
Cisco-IOS-XR-man-ems-oper.yang	Cisco-IOS-XR-bundlemgr-oper.yang
Cisco-IOS-XR-mpls-isd-oper.yang	Cisco-IOS-XR-l2vpn-oper.yang
Cisco-IOS-XR-show-fpd-loc-ng-oper.yang	Cisco-IOS-XR-asr9k-qos-oper.yang
Cisco-IOS-XR-telemetry-model-driven-oper.yang	Cisco-IOS-XR-segment-routing-ms-oper.yang
Cisco-IOS-XR-shellutil-oper.yang	Cisco-IOS-XR-pfi-im-cmd-oper.yang
Cisco-IOS-XR-ip-iep-oper.yang	Cisco-IOS-XR-asic-errors-oper.yang
Cisco-IOS-XR-cdp-oper.yang	Cisco-IOS-XR-lib-keychain-oper.yang
Cisco-IOS-XR-ip-sbfd-oper.yang	Cisco-IOS-XR-sdr-invmgr-oper.yang
Cisco-IOS-XR-tty-management-cmd-oper.yang	Cisco-IOS-XR-ipv4-ospf-oper.yang
Cisco-IOS-XR-upgrade-fpd-oper.yang	Cisco-IOS-XR-pfm-oper.yang
Cisco-IOS-XR-crypto-macsec-secy-oper.yang	Cisco-IOS-XR-config-valid-ccv-oper.yang

Cisco-IOS-XR-ip-iarm-v6-oper.yang	Cisco-IOS-XR-ip-iarm-v4-oper.yang
Cisco-IOS-XR-ipv4-autorp-oper.yang	Cisco-IOS-XR-infra-statsd-oper.yang
Cisco-IOS-XR-pbr-vservice-ea-oper.yang	Cisco-IOS-XR-ipv4-vrrp-oper.yang
Cisco-IOS-XR-ip-domain-oper.yang	Cisco-IOS-XR-cmproxy-oper.yang
Cisco-IOS-XR-ipv4-io-oper.yang	Cisco-IOS-XR-crypto-ssh-oper.yang
Cisco-IOS-XR-ipv4-hsrp-oper.yang	Cisco-IOS-XR-controller-optics-oper.yang
Cisco-IOS-XR-freqsync-oper.yang	Cisco-IOS-XR-atm-vcm-oper.yang
Cisco-IOS-XR-aaa-diameter-oper.yang	Cisco-IOS-XR-dnx-driver-fabric-plane-oper.yang
Cisco-IOS-XR-ip-tcp-oper.yang	Cisco-IOS-XR-asr9k-lc-fca-oper.yang
Cisco-IOS-XR-drivers-media-eth-oper.yang	Cisco-IOS-XR-mpls-vpn-oper.yang
Cisco-IOS-XR-infra-policymgr-oper.yang	Cisco-IOS-XR-asr9k-sc-envmon-oper.yang
Cisco-IOS-XR-fretta-bcm-dpa-hw-resources-oper.yang	Cisco-IOS-XR-es-acl-oper.yang
Cisco-IOS-XR-subscriber-ipsub-oper.yang	Cisco-IOS-XR-evpn-oper.yang
Cisco-IOS-XR-infra-rsi-oper.yang	Cisco-IOS-XR-rptiming-tmg-oper.yang
Cisco-IOS-XR-prm-server-oper.yang	Cisco-IOS-XR-ethernet-lldp-oper.yang
Cisco-IOS-XR-l2rib-oper.yang	Cisco-IOS-XR-ip-ntp-oper.yang
Cisco-IOS-XR-subscriber-pppoe-ma-oper.yang	Cisco-IOS-XR-mediasvr-linux-oper.yang
Cisco-IOS-XR-ocni-local-routing-oper.yang	Cisco-IOS-XR-ipv6-ma-oper.yang
Cisco-IOS-XR-reboot-history-oper.yang	Cisco-IOS-XR-infra-rmf-oper.yang
Cisco-IOS-XR-asr9k-lpts-oper.yang	Cisco-IOS-XR-infra-correlator-oper.yang
Cisco-IOS-XR-infra-serg-oper.yang	Cisco-IOS-XR-mpls-static-oper.yang
Cisco-IOS-XR-rgmgr-oper.yang	Cisco-IOS-XR-snmp-entitymib-oper.yang
Cisco-IOS-XR-ncs1k-mxp-headless-oper.yang	Cisco-IOS-XR-pbr-vservice-mgr-oper.yang
Cisco-IOS-XR-aaa-nacm-oper.yang	Cisco-IOS-XR-pfi-im-cmd-ctrlr-oper.yang
Cisco-IOS-XR-infra-rcmd-oper.yang	Cisco-IOS-XR-fretta-bcm-dpa-resources-oper.yang
Cisco-IOS-XR-crypto-macsec-mka-oper.yang	Cisco-IOS-XR-macsec-ctrlr-oper.yang
Cisco-IOS-XR-tunnel-vpdn-oper.yang	Cisco-IOS-XR-ipv6-nd-oper.yang
Cisco-IOS-XR-ipv4-dhcpd-oper.yang	Cisco-IOS-XR-tunnel-l2tun-oper.yang
Cisco-IOS-XR-ip-rip-oper.yang	Cisco-IOS-XR-infra-dumper-exception-oper.yang
Cisco-IOS-XR-ncs1001-otdr-oper.yang	Cisco-IOS-XR-syncc-oper.yang
Cisco-IOS-XR-asr9k-asic-errors-oper.yang	Cisco-IOS-XR-dnx-driver-oper.yang
Cisco-IOS-XR-pmengine-oper.yang	Cisco-IOS-XR-ncs1k-macsec-ea-oper.yang
Cisco-IOS-XR-linux-os-reboot-history-oper.yang	Cisco-IOS-XR-fretta-bcm-dpa-drop-stats-oper.yang

Cisco-IOS-XR-ppp-ea-oper.yang	Cisco-IOS-XR-infra-sla-oper.yang
Cisco-IOS-XR-asr9k-ntp-pd-oper.yang	Cisco-IOS-XR-ncs1001-ots-oper.yang
Cisco-IOS-XR-ipv4-igmp-oper.yang	Cisco-IOS-XR-nto-misc-shmem-oper.yang
Cisco-IOS-XR-ipv4-bgp-oc-oper.yang	Cisco-IOS-XR-ip-rib-ipv4-oper.yang
Cisco-IOS-XR-ip-pfilter-oper.yang	Cisco-IOS-XR-ipv4-pim-oper.yang
Cisco-IOS-XR-lpts-pre-ifib-oper.yang	Cisco-IOS-XR-pppoe-ea-oper.yang
Cisco-IOS-XR-ipv6-ospfv3-oper.yang	Cisco-IOS-XR-infra-syslog-oper.yang
Cisco-IOS-XR-asr9k-netflow-oper.yang	Cisco-IOS-XR-crypto-sam-oper.yang
Cisco-IOS-XR-infra-xtc-oper.yang	Cisco-IOS-XR-Ethernet-SPAN-oper.yang
Cisco-IOS-XR-sysdb-oper.yang	Cisco-IOS-XR-lpts-ifib-oper.yang
Cisco-IOS-XR-lib-mpp-oper.yang	Cisco-IOS-XR-ethernet-link-oam-oper.yang
Cisco-IOS-XR-infra-xtc-agent-oper.yang	Cisco-IOS-XR-mpls-ldp-oper.yang
Cisco-IOS-XR-ip-rib-ipv6-oper.yang	Cisco-IOS-XR-tty-management-oper.yang
Cisco-IOS-XR-rptiming-dti-oper.yang	Cisco-IOS-XR-lmp-oper.yang
Cisco-IOS-XR-wd-oper.yang	Cisco-IOS-XR-nto-misc-shprocmem-oper.yang
Cisco-IOS-XR-man-xml-ttyagent-oper.yang	Cisco-IOS-XR-procmem-oper.yang
Cisco-IOS-XR-ip-daps-oper.yang	Cisco-IOS-XR-Subscriber-infra-subdb-oper.yang
Cisco-IOS-XR-spirit-install-instmgr-oper.yang	Cisco-IOS-XR-asr9k-np-oper.yang
Cisco-IOS-XR-fretta-grid-svr-oper.yang	Cisco-IOS-XR-ntp-oper.yang
Cisco-IOS-XR-clns-isis-oper.yang	Cisco-IOS-XR-tunnel-nve-oper.yang
Cisco-IOS-XR-ipv4-bgp-oper.yang	Cisco-IOS-XR-ocni-oper.yang
Cisco-IOS-XR-ipv4-ma-oper.yang	Cisco-IOS-XR-ncs6k-acl-oper.yang
Cisco-IOS-XR-l2-eth-infra-oper.yang	Cisco-IOS-XR-manageability-object-tracking-oper.yang
Cisco-IOS-XR-plat-chas-invmgr-oper.yang	Cisco-IOS-XR-ocni-intfbase-oper.yang
Cisco-IOS-XR-dwdm-ui-oper.yang	Cisco-IOS-XR-infra-tc-oper.yang
Cisco-IOS-XR-policy-repository-oper.yang	Cisco-IOS-XR-subscriber-session-mon-oper.yang
Cisco-IOS-XR-ipv6-new-dhcpv6d-oper.yang	Cisco-IOS-XR-ip-udp-oper.yang
Cisco-IOS-XR-subscriber-srg-oper.yang	Cisco-IOS-XR-ipv6-acl-oper.yang
Cisco-IOS-XR-manageability-perfmgmt-oper.yang	Cisco-IOS-XR-crypto-macsec-pl-oper.yang
Cisco-IOS-XR-dnx-port-mapper-oper.yang	Cisco-IOS-XR-aaa-tacacs-oper.yang
Cisco-IOS-XR-mpls-te-oper.yang	Cisco-IOS-XR-man-ipsla-oper.yang
Cisco-IOS-XR-nto-misc-oper.yang	Cisco-IOS-XR-invmgr-oper.yang
Cisco-IOS-XR-ppp-ma-oper.yang	Cisco-IOS-XR-ipv4-arp-oper.yang

Cisco-IOS-XR-config-cfgmgr-exec-oper.yang	Cisco-IOS-XR-aaa-locald-oper.yang
Cisco-IOS-XR-perf-meas-oper.yang	Cisco-IOS-XR-ha-eem-policy-oper.yang
Cisco-IOS-XR-snmp-agent-oper.yang	Cisco-IOS-XR-ascii-ltrace-oper.yang
Cisco-IOS-XR-asr9k-lc-ethctrl-oper.yang	Cisco-IOS-XR-skp-qos-oper.yang
Cisco-IOS-XR-ifmgr-oper.yang	Cisco-IOS-XR-flowspec-oper.yang
Cisco-IOS-XR-iedge4710-oper.yang	Cisco-IOS-XR-icpe-sdaccp-oper.yang
Cisco-IOS-XR-controller-otu-oper.yang	Cisco-IOS-XR-fretta-bcm-dpa-npu-stats-oper.yang
Cisco-IOS-XR-subscriber-accounting-oper.yang	Cisco-IOS-XR-alarmgr-server-oper.yang
Cisco-IOS-XR-ncs5500-qos-oper.yang	Cisco-IOS-XR-fia-internal-tcam-oper.yang
Cisco-IOS-XR-skywarp-netflow-oper.yang	Cisco-IOS-XR-tty-server-oper.yang
Cisco-IOS-XR-ncs1k-mxp-ldp-oper.yang	Cisco-IOS-XR-qos-ma-oper.yang
Cisco-IOS-XR-fib-common-oper.yang	Cisco-IOS-XR-aaa-protocol-radius-oper.yang
Cisco-IOS-XR-dnx-netflow-oper.yang	Cisco-IOS-XR-platform-pifib-oper.yang
Cisco-IOS-XR-lpts-pa-oper.yang	Cisco-IOS-XR-asr9k-fsi-oper.yang
Cisco-IOS-XR-ncs1k-mxp-oper.yang	Cisco-IOS-XR-ncs5500-coherent-node-oper.yang
Cisco-IOS-XR-asr9k-sc-invmgr-oper.yang	Cisco-IOS-XR-snmp-ifmib-oper.yang
Cisco-IOS-XR-ptp-pd-oper.yang	Cisco-IOS-XR-ip-mobileip-oper.yang
Cisco-IOS-XR-ethernet-cfm-oper.yang	Cisco-IOS-XR-wdsysmon-fd-oper.yang
Cisco-IOS-XR-pbr-oper.yang	Cisco-IOS-XR-infra-objmgr-oper.yang
Cisco-IOS-XR-ip-rsvp-oper.yang	Cisco-IOS-XR-ipv6-io-oper.yang
Cisco-IOS-XR-terminal-device-oper.yang	Cisco-IOS-XR-plat-chas-invmgr-ng-oper.yang
Cisco-IOS-XR-mpls-oam-oper.yang	Cisco-IOS-XR-ncs5500-coherent-portmode-oper.yang
Cisco-IOS-XR-sse-span-oper.yang	Cisco-IOS-XR-infra-dumper-oper.yang
Cisco-IOS-XR-asr9k-sc-diag-oper.yang	Cisco-IOS-XR-mpls-io-oper.yang