New and Changed Feature Information

This section lists all the new and changed features for the Programmability Configuration Guide.

- New and Changed Programmability Features, on page 2

New and Changed Programmability Features

Feature	Description	Changed in Release	Where Documented
Support to poll specific processes to stream telemetry data.		Release 7.1.2	Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented

	Introduced `Cisco-IOS-XR-wdsysmon-fd-proc-oper.yang` data model with process keys to poll specific processes and stream telemetry data.
	NETCONF Request:
	`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
	`<get>`
	`<filter>`
	`<process-monitoring xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-wdsysmon-fd-proc-oper">`
	`<nodes>`
	`<node>`
	`<node-name>0/RP0/CPU0</node-name>`
	`<process-name>`
	`<proc-cpu-utilizations>`
	`<proc-cpu-utilization>`
	`<process-name>`
	`dumper`
	`</process-name>`
	`</proc-cpu-utilization>`
	`</proc-cpu-utilizations>`
	`</process-name>`
	`</node>`
	`</nodes>`
	`</process-monitoring>`
	`</filter>`
	`</get>`
	`</rpc>`
	NETCONF Response:
	`<?xml version="1.0"?>`
	`<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
	`<data>`
	`<process-monitoring xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-wdsysmon-fd-proc-oper">`
	`<nodes>`
	`<node>`
	`<node-name>0/RP0/CPU0</node-name>`
	`<process-name>`		

Feature	Description	Changed in Release	Where Documented
	<proc-cpu-utilizations> <proc-cpu-utilization> <process-name>dumper</process-name> <total-cpu-one-minute>0</total-cpu-one-minute> <total-cpu-five-minute>0</total-cpu-five-minute> <total-cpu-fifteen-minute>0</total-cpu-fifteen-minute> <process-cpu> <process-name>dumper</process-name> <process-id>3572</process-id> <process-cpu-one-minute>0</process-cpu-one-minute> <process-cpu-five-minute>0</process-cpu-five-minute> <process-cpu-fifteen-minute>0</process-cpu-fifteen-minute> <thread-cpu> <thread-name>lwm_service_thr</thread-name> <thread-id>3597</thread-id> <process-cpu-one-minute>0</process-cpu-one-minute> <process-cpu-five-minute>0</process-cpu-five-minute> <process-cpu-fifteen-minute>0</process-cpu-fifteen-minute> </thread-cpu> <thread-cpu> <thread-name>qsm_service_thr</thread-name> <thread-id>3602</thread-id> <process-cpu-one-minute>0</process-cpu-one-minute> <process-cpu-five-minute>0</process-cpu-five-minute>		

Feature	Description	Changed in Release	Where Documented
	<process-cpu-fifteen-minute>0</process-cpu-fifteen-minute> </thread-cpu> <thread-cpu> <thread-name>dumper</thread-name> <thread-id>3605</thread-id> <process-cpu-one-minute>0</process-cpu-one-minute> <process-cpu-five-minute>0</process-cpu-five-minute> <process-cpu-fifteen-minute>0</process-cpu-fifteen-minute> </thread-cpu> <thread-cpu> <thread-name>dumper</thread-name> <thread-id>3611</thread-id> <process-cpu-one-minute>0</process-cpu-one-minute> <process-cpu-five-minute>0</process-cpu-five-minute> <process-cpu-fifteen-minute>0</process-cpu-fifteen-minute> </thread-cpu> <thread-cpu> <thread-name>dumper</thread-name> <thread-id>3613</thread-id> <process-cpu-one-minute>0</process-cpu-one-minute> <process-cpu-five-minute>0</process-cpu-five-minute> <process-cpu-fifteen-minute>0</process-cpu-fifteen-minute> </thread-cpu> <thread-cpu> <thread-name>evm_signal_thre</thread-name> <thread-id>4462</thread-id> <process-cpu-one-minute>0</process-cpu-one-minute>		

Feature	Description	Changed in Release	Where Documented

	`<process-cpu-five-minute>0</process-cpu-five-minute>`
	`<process-cpu-fifteen-minute>0</process-cpu-fifteen-minute>`
	`</thread-cpu>`
	`</process-cpu>`
	`</proc-cpu-utilization>`
	`</proc-cpu-utilizations>`
	`</process-name>`
	`</node>`
	`</nodes>`
	`</process-monitoring>`
	`</data>`
	`</rpc-reply>`		

OCNI BGP advertised prefix	OCNI BGP data model has global, neighbor and peer-group containers. Under the neighbor container are the Address Family Identifier (AFI) state-related leaves. Three leaves (sent, installed and advertised prefix count) are related to the prefixes received from and advertised to the neighbor specific to that AFI. The sent count is defined as how many prefixes are sent to the neighbor for a specific AFI. In the existing implementation, the cumulative count may not display accurate value owing to various conditions such as route-refresh out, policy changes, route-target changes, configuration changes like nexthop-self to name a few. In this release, BGP walks through the prefix table calculating exactly how many prefixes are advertised for each neighbor. The results are collected in an array, converted and sent to MDT.	Release 7.1.2	Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented
Generic Routing Encapsulation (GRE) Tunnel configuration support added to oc-interfaces data model	oc-interfaces data model is used to manage network interfaces and subinterfaces. In this release, oc-interfaces data model supports additional sensor paths for tunnel configuration and state parameters:: • `src` • `dst` • `ttl` (time-to-live or hop limit • `gre-key`	Release 7.1.2	*Drive Network Automation Using Programmable YANG Data Models* chapter YANG Data Model Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented
Table connection support in `openconfig-network instance` (oc-ni) data model		Release 7.1.2	Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented

	The oc-ni data model is defined by OpenConfig community. This model defines the network instance concept to model Layer 3 and Layer 2 network instances applicable for services such as L3VPN, L2VPN, and EVPN.
	The table-conection container in the oc-ni data model contains policies that dictate how routing information base (RIB) or forwarding information base (FIB) entries are propagated between routing tables.
	The leaf list consists of a list of connections between pairs of routing or forwarding tables, the leaking of entries between which is specified by the import policy. A connection connecting a source table to a destination table implies that routes that match the policy specified for the connection are available for the destination protocol to advertise, or match within its policies. It shows the configuration and state parameters that relate to the connection between tables.
	The following CLIs corresponds to the configuration of oc-ni nodes:
	`Router static address-family ipv4 unicast`
	`Router static address-family ipv6 unicast`
	`<prefix> <interface> <next-hop> tag <tag> metric <metric>`
	The following CLIs corresponds to the state of oc-ni nodes:
	`show running-config`
	`show running-config router static`
	`show running-config router static address-family ipv4`		

Feature	Description	Changed in Release	Where Documented
	`unicast`		
Support for `openconfig-system` data model	Support for oc-system data model to manage system-wide services and functions on network devices. Use this data model to configure and retrieve operational state data of the system. This model includes the following components: • openconfig-aaa (includes AAA and TACACS) • openconfig-system-logging • openconfig-alarms • openconfig-system-management • openconfig-system-terminal • openconfig-messages	Release 7.1.15	Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented
Revision of oc-interfaces.yang model from version 1.0.2 to version 2.4.0	oc-interfaces data model is used to manage network interfaces and subinterfaces. This module also defines types and groupings for other models to create references on interfaces. In this release, oc-interfaces data model supports additional sensor paths: • Configuration parameter: • `loopback-mode` for interfaces • State parameter: • `loopback-mode` for interfaces • `logical` leaf nodes for interfaces and sub interfaces • Counters: • `in-pkts` • `in-fcs-errors` • `carrier-transitions` • `out-pkts` • Data type and definition: • `last-change` • `last-clear`	Release 7.1.1	*Drive Network Automation Using Programmable YANG Data Models* chapter YANG Data Model Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented
gRPC Network Operations Interface (gNOI)	gNOI defines a set of gRPC-based microservices for executing operational commands on network devices. Extensible Manageability Services (EMS) gNOI is the Cisco IOS XR implementation of gNOI. gNOI supports for the following remote procedure calls (RPCs): • System • Ping • Traceroute • Time • SwitchControlProcessor • File • Stat • Put • TransferToRemote • Cert • Rotate • Install • GetCertificates • RevokeCertificates • CanGenerateCSR	Release 7.1.1	*Use gRPC Protocol to Define Network Operations with Data Models* chapter gRPC Network Operations Interface

Feature	Description	Changed in Release	Where Documented
Support for `oc-bfd` data model	The OpenConfig -Bidirectional Forwarding Detection (oc-bfd) data model defines the BFD protocol in multi-vendor environment to configure and get operational state data for the BFD protocol. **Note** The `oc-bfd` data model does not support `micro-bfd` container, and telemetry data push or Event-driven telemetry (EDT).	Release 7.1.1	*Drive Network Automation Using Programmable YANG Data Models* chapter YANG Data Model Obtain this data model from Github repository.
Support for `mldp` data model	The native Multicast Label Distribution Protocol (mldp) model defines configuration and operational state data for the MLDP protocol.	Release 7.1.1	*Drive Network Automation Using Programmable YANG Data Models* chapter YANG Data Model Obtain this data model from Github repository.
Revised `oc-isis` data model	The OpenConfig - Integrated Intermediate System-to-Intermediate System (oc-isis) model defined by the OC community. In this release, support is provided for additional paths in oc-isis data model.	Release 7.1.1	Obtain this data model from Github repository.
Revised `oc-policy` data model	The OpenConfig-Policy (oc-policy) data model defined by the OC community contains general data definitions for use in routing policy. It can be imported by modules that contain protocol-specific policy conditions and actions. In this release, support is extended for additional paths in the oc-policy data model with a revised version 2.0.1 to version 3.1.0. For information about limitations for this feature, see the *Note* section after this table.	Release 7.1.1	Obtain this data model from Github repository.

Feature	Description	Changed in Release	Where Documented

Power ON and Power OFF a line card using `oc-platform` data model	Enhancement to support the ability to power ON and power OFF a line card using oc-platform data model.	Release 7.1.1	Obtain this data model from Github repository.
			The relevant model paths are:
			`/components/component/linecard/config/power-admin-state`
			`/components/component/linecard/state/power-admin-state`

Stream telemetry data using `openconfig-platform` data model	Streaming data related to the underlying characteristics of the device including the operational state or configuration of that device using openconfig-platform data model.	Release 7.1.1	Obtain this data model from Github repository.
Revised gNMI features	Enhancement of gNMI specification to include updates from version 0.4.0 to version 0.6.0. Support is extended for the following gNMI features: • gNMI support for multiple client roles and primary arbitration • Path Target • gNMI service registration with the gRPC reflection service to allow clients to determine that gNMI is available on the target	Release 7.1.1	*Use gRPC Protocol to Define Network Operations with Data Models* chapter gRPC Network Management Interface For more information about the enhanced features, see gRPC Network Management Interface (gNMI) in Github repository.