

# Open ROADM MSA Device White paper



V1.01 10/25/17

Please download the latest version on <http://OpenROADM.org>

# 1 TABLE OF CONTENTS

---

2	High level Device description .....	6
2.1	Physical design .....	6
2.2	Functional design .....	6
2.2.1	ROADM.....	6
2.2.2	Transponder .....	8
2.2.3	Pluggable Optics.....	9
2.2.4	YANG Data Model .....	9
2.2.5	Open ROADM controller .....	10
2.3	Open Interfaces.....	10
2.3.1	W .....	10
2.3.2	Wr.....	10
2.3.3	MW.....	10
2.3.4	OSC.....	10
3	Node description, installation and discovery.....	11
3.1	File Operations and structure .....	11
3.1.1	Local device file structure .....	11
3.1.2	SFTP specification.....	11
3.1.3	File transfer (upload).....	11
3.1.4	File transfer (download).....	12
3.1.5	Retrieve file list .....	12
3.1.6	Delete file .....	13
3.1.7	File operation notifications .....	13
3.2	Data Communication Network (DCN) .....	13
3.3	NETCONF Protocol .....	15
3.3.1	SSHv2 support .....	15
3.3.2	Notification support.....	16
3.3.3	NETCONF monitoring .....	16
3.3.4	Change notifications. ....	16
3.3.5	YANG modeling .....	16

3.4	Open ROADM Discovery and Commissioning.....	16
3.4.1	Pre-planned device template.....	18
3.4.2	System Initialization and auto-provisioning.....	18
3.4.3	DHCP .....	19
3.4.4	Controller discovery of new IP address assignment .....	19
3.4.5	Initial discovery of the network element.....	19
3.4.6	Correlation of the planned template and the temporary discovered node .....	19
3.4.7	Node commissioning.....	20
3.4.8	Permanent Node Discovery .....	22
3.5	OMS Link Planning and Discovery.....	22
4	Device software, firmware and database .....	23
4.1	Software upgrade.....	23
4.1.1	Manifest file .....	23
4.2	software downgrade.....	23
4.3	Firmware upgrade.....	23
4.4	database operations .....	23
4.4.1	database backup .....	23
4.4.2	database restore .....	23
4.4.3	database change notifications .....	23
4.5	syslog.....	23
5	Notifications.....	24
5.1	PM .....	24
5.2	Alarms .....	35
5.3	Threshold Crossing Alarms (TCAs) .....	45
5.4	Operational states and database change notifications .....	45
6	Device specific topics .....	45
6.1	ROADM.....	45
6.1.1	OTDR scan .....	45
6.1.2	Interfaces .....	45
6.2	Transponder.....	45
6.2.1	Interfaces .....	45
6.3	Pluggable.....	45
7	Provisioning action use cases.....	45

7.1	Connection management .....	46
7.1.1	Xponder interfaces creation (TX) .....	46
7.1.2	Add-Link creation .....	49
7.1.3	Express link creation .....	51
7.1.4	Drop-Link creation .....	54
7.1.5	Xponder interfaces creation (RX) .....	56
7.1.6	Xponder interfaces deletion (TX/RX) .....	59
7.1.7	Add-link deletion .....	60
7.1.8	Express link deletion .....	61
7.1.9	Drop link deletion.....	62
7.1.10	Note on State Models .....	63
7.2	incremental hardware .....	63
7.2.1	Degree Growth.....	63
7.2.2	SRG Growth.....	71
7.2.3	Xponder Growth.....	76
8	Maintenance action use cases .....	80
8.1	Reset/Restart at system or circuit pack level.....	80
8.2	Set OTU/ODU TTI fields (SAPI/DAPI message) for connectivity management .....	80
8.3	Operate/Release loopback on an interface .....	80
8.4	Generate/Collect debug dump file(s) for in-depth failure analysis .....	80
8.5	Restore remote communications to the device from corrupted or inconsistent configuration	80
8.6	Re-sync with northbound controller/applications after extended period of communication loss	81
8.1	Missing topics (to be addressed in future version of Whitepaper) .....	81
9	Appendix A: Manifest file for software download and database operations.....	82
9.1	Introduction .....	82
9.1.1	Software Download .....	82
9.1.2	Database Backup.....	83
9.1.3	Database Restore .....	83
9.2	Manifest File YANG module .....	84
9.2.1	Tree View .....	84
9.2.2	YANG module (this module will be finalized and provided as part of Open ROADM Version 2)	86
9.3	Sample Manifest Files .....	98

9.3.1	Example Software Download File .....	98
9.3.2	Example Database Backup Manifest File .....	99
9.3.3	Example Database Restore Manifest File .....	99

## 2 HIGH LEVEL DEVICE DESCRIPTION

---

### 2.1 PHYSICAL DESIGN

The physical design for Open ROADM devices is not specified in the MSA. The form factor (width, depth, shelf), power supply (AC/DC) or standards met (such as NEBS3) are up to the manufacturer or Network operator. The MSA only covers the functional aspects, as well as optical interoperability of the single- and multi-wave interface.

Since no function of a craft-interface terminal has been specified, a manual factory-reset mechanism (such as a physical button) is desired to get the network element back to the factory reset state (same state as before power was switched on the first time). Obviously, such reset mechanism needs to be hidden and clearly labeled that no accidental reset is triggered.

### 2.2 FUNCTIONAL DESIGN

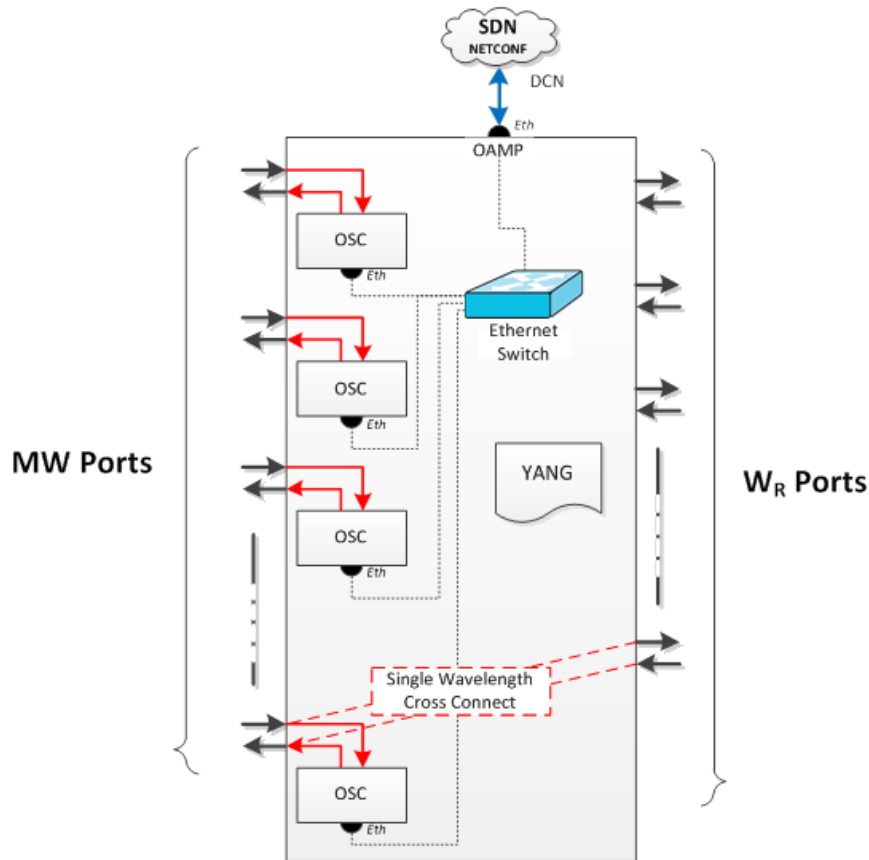
#### 2.2.1 ROADM

The Open ROADM MSA defines a ROADM device capable of providing colorless and directionless add/drop functionality. This means that a ROADM site can add/drop any wavelength at any port and connect that wavelength to any direction in a ROADM node<sup>1</sup> from the local transponder. The MSA does not define implementation details such as the number of degrees, form-factor, etc. The MSA defines:

- API using NETCONF interface with a YANG-based data model that abstracts the management, control and provisioning of multi-vendor ROADM devices
- Multi-wave (MW) interface which defines the optical specifications for the multi-wave DWDM interface between line degrees of the ROADM devices
  - The MW interface includes a GE OSC that provides LLDP-based topology information, support laser safety and provide MCN/DCN reach-through to remote ROADM nodes. Otherwise, there is no ROADM node to ROADM node communication and optical equalization is handled off-box by the controller.
- Single-wave (Wr) interface which defines the optical specifications for the add/drop ports of the ROADM devices

---

<sup>1</sup> The term “device”, “node” and “NE” are used interchangeably within this document.



Please refer to the Open ROADM MSA specification found at [openroadm.org](http://openroadm.org) for the C/D ROADM specifications:<sup>2</sup>

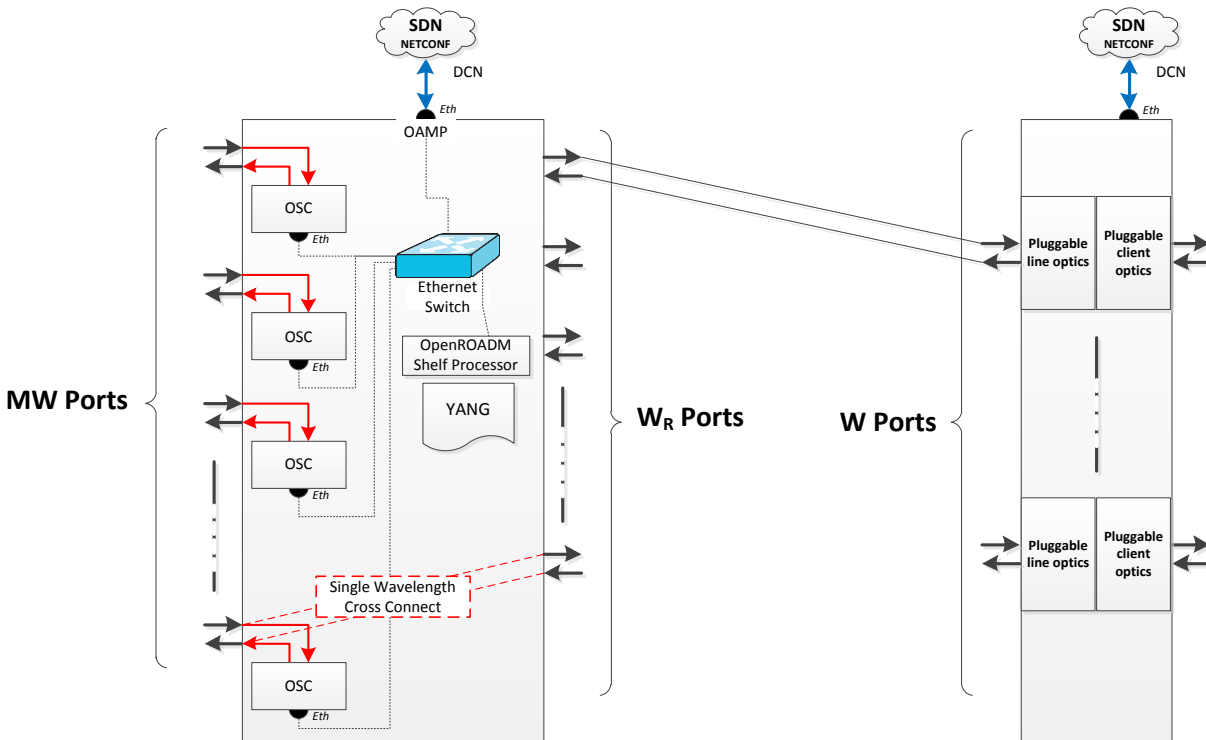
- General specs in the "Common" tab
- Physical specs defined in the "Physical spec" tab
- Line degree to line degree optical specs defined in the "MW-MW" tab
- Line degree to drop port and add port to line degree optical specs defined in the "MW-Wr" tab
- Local control specs defined in the "Local Control" tab
- Supported alarms defined in the "Alarms" tab
- Supported PMs defined in the "PMs" tab
- Ethernet OSC optical specs defined in the "OSC-Optical Line Port" tab
- Ethernet OSC functionality defined in the "OSC Overview" tab
- Automatic line shutoff functionality defined in the "Laser Safety" tab
- OAMP functions including protocol stack and specs in the "OAMP Port" tab
- OpenDaylight mounting requirements in the "Device mounting" tab
- OTDR functionality defined in the "OTDR" tab

<sup>2</sup> The current version of the MSA specification as of the writing of this document is 20170523-Open-ROADM-MSA-specification-ver-1-07.xlsx.

### 2.2.2 Transponder

The Open ROADM MSA defines a Transponder device capable of mapping a single 100GE or OTU4 client signal into a 100G OTU4 DWDM signal for transport across an Open ROADM infrastructure. The MSA does not define implementation details such as the number of client/line ports, form-factor, etc. The MSA (refer to [openroadm.org](http://openroadm.org)) defines:

- API using NETCONF interface with a YANG-based data model that abstracts the management, control and provisioning of multi-vendor Transponder devices
- Single-wave (W) interface which defines the optical specifications for the full C-band tunable DWDM optical line interface of the Transponder that connects to a  $W_R$  add/drop port on the ROADM device
  - Line-side pluggable type must be CFP-DCO, CFP2-ACO or CFP2-DCO with LC connectors
- Client interfaces/ports must be pluggable QSFP28 with LC connectors and support:
  - 100GBASE-R mapped into OPU4 using PCS codeword transparent Ethernet mapping
  - OTU4



Please refer to the Open ROADM MSA specification found at [openroadm.org](http://openroadm.org) for the Transponder specifications:

- General specs in the “Common” tab
- Optical specs defined in the “W Optical Spec” tab
- Support alarms defined in the “W ALM Spec” tab
- Support PMs defined in the “W PM Spec” tab
- Client functional specs defined in the “W TRPN functional” tab



- Physical specs defined in the “Physical spec” tab
- OAMP functions including protocol stack and specs in the “OAMP Port” tab
- OpenDaylight mounting requirements in the “Device mounting” tab

### 2.2.3 Pluggable Optics

The Open ROADM MSA defines the use of standards-based pluggable optics for the Transponder device. Client interfaces on the Transponder must use standard QSFP28 pluggable optics. Line interfaces on the Transponder must use standard CFP-DCO, CFP2-ACO or CFP2-DCO pluggable optics that conform to the W optical specification.

Otherwise, the Open ROADM MSA does not dictate nor preclude the use of pluggable optics to perform other functions within the ROADM or Transponder device.

### 2.2.4 YANG Data Model

The YANG data model consists of:

- Config / Operational data defines database objects which can be queried by the controller. Some database objects can be read/write (config), while others are read-only (operational). Examples include shelf commissioning data, wavelength connections, etc.
- Notifications for the purposes of reporting autonomous events to the controller. Examples include alarms, inventory changes, re-start, etc.
- Remote Procedure Calls (RPC) that do not effect a change in the device configuration data, e.g., get-connection-port-trail, file transfers, database backup, etc.

The Open ROADM device YANG data model defines the following objects to abstract the implementation of the ROADM and Transponder device:

- Info provide general node information including node name, IP address, etc.
- Shelves provide shelf information. A node can consist of one or more shelves.
- Circuit Packs represents a physical piece of equipment which contains a group of hardware functional blocks such as common equipment, cards, plug-in-units and/or pluggable optics.
  - The Ports container defines the ports associated with a circuit pack or pluggable optics and the associated port attributes
- Internal Links reflect the connectivity within each circuit pack. These objects are read only and report attributes of the circuit pack themselves.
- Physical Links reflect the connectivity between ports across different circuit packs. The controller pushes this data to the device and reflects the actual inter-card fiber/cabling.
- External Link objects are placeholders for data about the far end device. Data for these objects is pushed from the controller.
- Degrees define the grouping of circuit packs that form a line degree
- Shared Risk Groups define the grouping of circuit packs that form a colorless/ directionless add/drop bank
- Wavelength Map defines the wavelength channel number and wavelength map
- Connection Map is wavelength agnostic and reflects any connectivity restrictions / blocking in the device (not wavelength contention).
- Interfaces defines supported interface types and are associated with Port objects

### 2.2.5 Open ROADM controller

The Open ROADM architecture assumes the existence of an Open ROADM controller that controls the Open ROADM devices and provides device (inventory), network and service APIs to northbound OSS systems. The Open ROADM MSA does not specify the requirements, implementation or operation of an Open ROADM controller. The exception is the specification of the Open ROADM device, network and service models that would be used at the controller level.

Some controller functions and/or procedures may be provided within this document. These items should be taken as guidance or examples of a possible controller implementation to guide the development of an Open ROADM devices.

## 2.3 OPEN INTERFACES

### 2.3.1 W

The Single-Wavelength interface “W” specifies the interoperability of two transponders. The Optical specification outlines a minimum number of parameters for framing and bit ordering to enable the interoperability between different hardware manufacturers.

### 2.3.2 Wr

The Single-Wavelength port on the ROADM (also known as add/drop port) is called “Wr”. Here, a single-wavelength output “W” from an Xponder is plugged into the ROADM.

### 2.3.3 MW

The interoperability of different manufacturers ROADMs is guaranteed by defining the Multi-Wavelength interface “MW”. The Optical Specification sheet from OpenROADM.org contains the minimum interoperability specifications with some performance metric of the ROADM (see MW-MW and MW-Wr tabs).

### 2.3.4 OSC

The optical supervisory channel is part of the MW interoperability. The MSA has defined 100GBASE-LX interoperability and a simple Ethernet wayside channel (see OSC tabs in spreadsheet). Aside from the mandatory safety automatic power shutdown control loop (see Laser safety tab), no control loops are running between two adjacent ROADMs. All control loops are abstracted into the centralized controller. There are, however, a few local link control loops running on the device, such as transient control. See Local Control tab for further details.

MSA Version	Requirement
1.2	<ul style="list-style-type: none"><li>• <b>Single wavelength interface:</b> Only 100G coherent PM-QPSK has been specified with the Staircase Forward-Error Correction (FEC)</li><li>• <b>Multi-wavelength interface:</b> Fixed grid at 50GHz spacing with 96 wavelengths has been defined</li></ul>

## 3 NODE DESCRIPTION, INSTALLATION AND DISCOVERY

---

### 3.1 FILE OPERATIONS AND STRUCTURE

#### 3.1.1 Local device file structure

The file system on the Open ROADM device follows a flat structure with no subdirectories. When an operation generates multiple files, these would be tar'd then zip'd in a single file which is stored locally on the flat structure on the device.

The allocated space on the device shall be large enough to accommodate at least one copy of the files required to support operations requiring files exchange with the Open ROADM controller (ex: debug, syslogs, database for backup and restore, software loads for upgrade, OTDR trace, etc.)

#### 3.1.2 SFTP specification

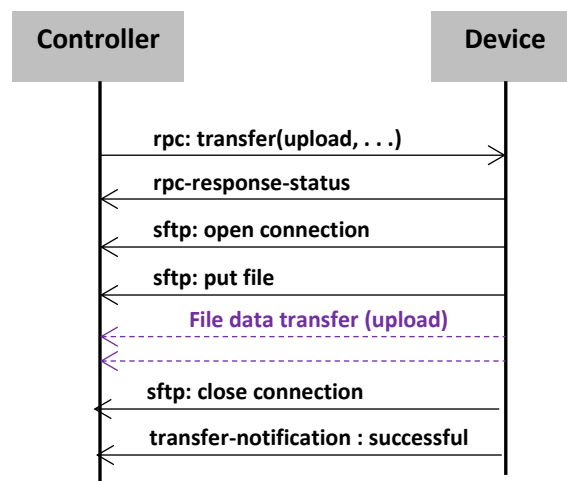
SFTP is used to exchange files between the Open ROADM controller and the devices; in such case the device is the SFTP client and the Open ROADM controller the SFTP server, in such scenario the device would initiate the SFTP connection to the server, a file transfer to the device would translate in a “get” operation triggered by the device and a file transfer from the device would translate in a “put” operation triggered by the device. The SFTP server port can be specified as an optional parameter in the URL of the transfer RPC, if a port is not specified the device shall use default port 22.

#### 3.1.3 File transfer (upload)

rpc: transfer

- action: upload (from device to Open ROADM Controller)
- local-file-path: file on the device
- remote-file-path: URI of file on the Open ROADM Controller including credentials

It was agreed that the operation be asynchronous with rpc-response-status is sent when the request is accepted

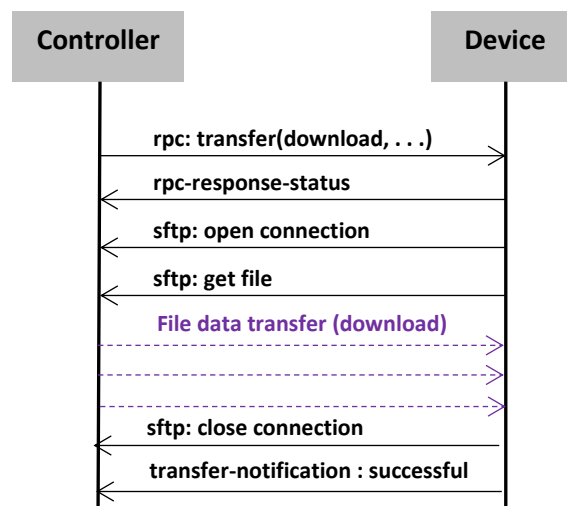


### 3.1.4 File transfer (download)

rpc: transfer

- [input] action: download (from Open ROADM Controller to device)
- [input] local-file-path: file on the device
- [input] remote-file-path: URI of file on the Open ROADM Controller including credentials

It was agreed that the operation be asynchronous with rpc-response-status is sent when the request is accepted.



### 3.1.5 Retrieve file list

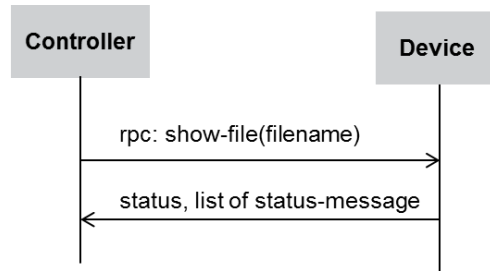
rpc: show-file

- [Input] filename: file to be listed (\* is allowed as wild-card)
- [Output] status: Successful or Failed
- [Output] list of status-message, each one contains one filename

The file list operation is only applicable to the content of the flat Open ROADM directory structure. In terms of operations it was agreed that:

- A "\*" or blank input would list all the files in the Open ROADM flat structure
- Only one filename can be specified as an input
- Partial wildcarding is not supported (ex: abc\*)
- An empty list output would be represented by a success status with no file list

The MSA members will consider enhancements in a future release to show additional file information (ex: file size and timestamp).



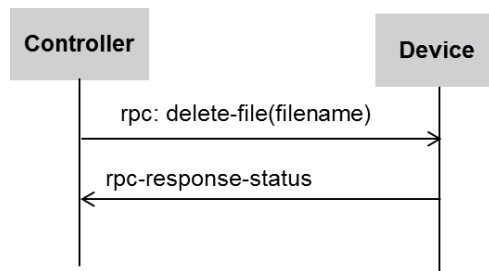
### 3.1.6 Delete file

rpc: delete-file

- [Input] filename: name of file to delete
- [Output] rpc-response-status

The file delete operation is only applicable to the content of the flat Open ROADM directory structure. In terms of operations it was agreed that:

- Input wildcarding (ex: “\*”) is not supported
- Only one filename can be specified as an input

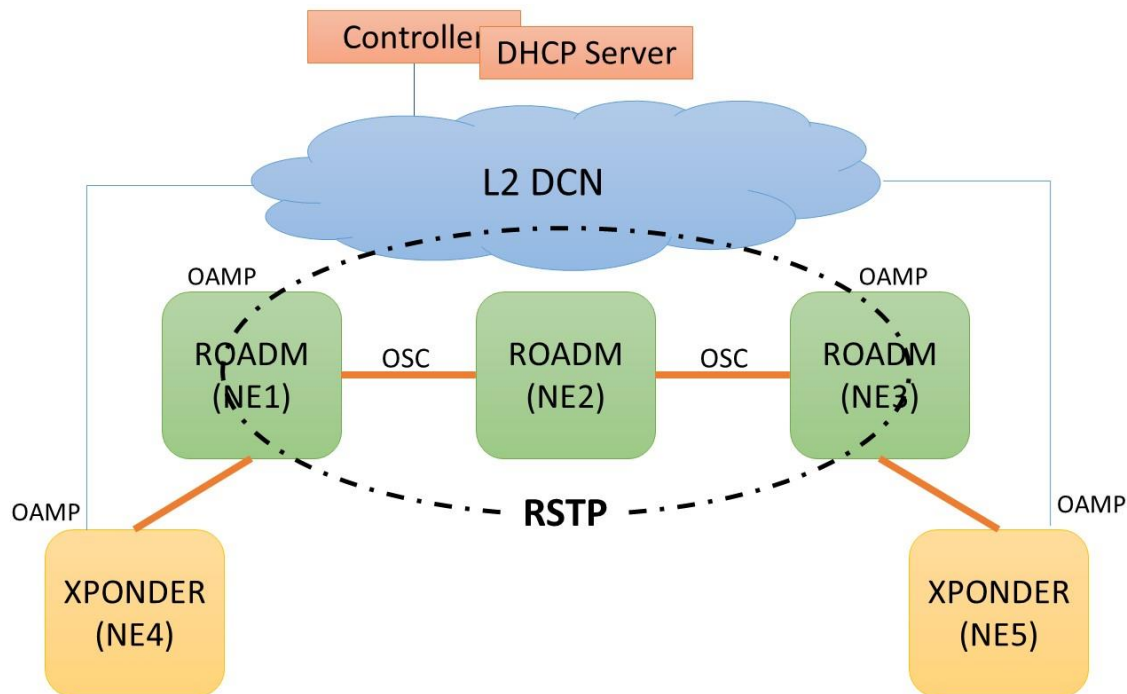


### 3.1.7 File operation notifications

Full support for transient notifications were not provided in the Open ROADM Version 1.2.1 YANG models. As a short-term workaround, file operation transient notifications are sent as alarms with severity set to 'indeterminate'. These notifications would not have a corresponding clear notification unlike a normal alarm. Therefore, in Open ROADM Version 1.2.1, traditional alarms should not be raised with the 'indeterminate' status.

## 3.2 DATA COMMUNICATION NETWORK (DCN)

Figure below shows the overview of the Open ROADM DCN architecture. Each device acts as a layer 2 bridge and runs Rapid Spanning Tree (RSTP) to provide loop free topology.



For the ROADM NE:

- Each ROADM NE is running in L2 bridging mode
- NE1 and NE3 have OAMP connection to DCN while NE2 is only reachable via OSC
- Single bridge with Bridge ports { OAMP, OSC1, OSC2, ...}
- RSTP is enabled on OAMP and OSC ports to provide loop free topology
- NE1 and NE3 can be reached via OAMP if RSTP has not blocked that port otherwise reachable via OSC.
- NE2 is reachable via OSC
- Customer L2 DCN also participate in RSTP
- 

For Transponder Node:

- Has single management port OAMP.
- Acts as a host since there is only single management port.
- Does not need to participate in the RSTP exchange as the transponder node is a host

When a ROADM or transponder NE is powered up, it runs IPv4 and IPv6 DHCP clients and gets an IP address from the DHCP server sitting on the same LAN. The IP address may be either IPv4 or IPv6 depending on the operator's DCN configuration. This IP address is called temporary IP address. When DHCP server allocates temporary IP address for a NE, then Controller is notified for this new IP address allocation. Now controller can login to this NE using temporary address and can give the permanent IP address as per the carrier planning.

Provisioned IP address on the NE can be IPV6 or IPV4. NE also allows to provision the default gateway. The IP address, prefix length and default gateway should be specified in the same edit config operation.

### 3.3 NETCONF PROTOCOL

The Open ROADM network element is managed using the NETCONF protocol [RFC 6241] on TCP port 830. The default username and password for accessing a NETCONF network element is openroadm/openroadm.

The Open ROADM network element advertises its capabilities in the NETCONF Hello message. The Hello message provides an indication of support for standard features defined in NETCONF RFCs as well as support for specific namespaces.

Configuration data can be written directly to the running configuration datastore or written to the candidate configuration datastore with a subsequent commit to push the configuration to the running datastore. The specific mechanism supported by a Network Element is advertised in the Hello message. A vendor can support writing to the running configuration only, writing to the candidate configuration only, or support writing to both the running and candidate configurations.

NETCONF requires the support for subtree filtering on NETCONF messages. Network elements may also support XPATH filtering. An Open ROADM network element indicates its support for XPATH in the Hello message.

Operations on the network element should be idempotent when the NETCONF "merge" operation is used. If the data in the merge operation is the same as what already exists on the network element (even if it specifies only a partial set of the total attributes on the NE), the NE will accept the command with no changes. No database change notification would be sent under this case.

#### 3.3.1 SSHv2 support

The NETCONF protocol runs over SSHv2 for security. It is recommended that implementations follow RFC 6242 which is the latest specification of NETCONF over SSH. RFC 6242 provides the procedure for interoperability with NETCONF implementations that support the older NETCONF over SSH described in RFC 4742.

Note: In RFC 4742, "[>]]]" was used as the NETCONF message delimiter. But an issue was identified where this character string could appear in the NETCONF body causing parsing issues. RFC 6264 modifies the message format to support message chunks with explicit message size identifiers to overcome this issue.

If multiple NETCONF sessions are established to a network element, those sessions should be established over separate SSH tunnels.

The Open ROADM network element uses the password authentication method for SSH. Once authenticated, the controller will request to open a channel of type "session" and invoke the "netconf" subsystem.

Devices may support IDLE timeout to clean up inactive sessions. In such case, keep alive messages are required to maintain connectivity. For example, an Open ROADM controller may send periodic retrieves of the info container using a ping mechanism.

### 3.3.2 Notification support

Open ROADM network elements must support NETCONF event notifications [RFC 5277]. Implementations may choose to output the Open ROADM notifications on the optional OPENROADM stream. If this stream is used, only Open ROADM notifications should be exposed on this stream. If the OPENROADM stream is not supported then implementations must output the Open ROADM notifications on the default NETCONF stream.

Note that there is no guarantee that only Open ROADM notifications would appear on the default NETCONF stream.

A controller should retrieve the list of streams supported by an Open ROADM network element. If the NE supports the OPENROADM stream, the controller should subscribe to that stream. Otherwise it should subscribe to the NETCONF (default) stream.

Open ROADM devices must support the interleave option that allows both notifications and RPCs in the same session.

### 3.3.3 NETCONF monitoring

Open ROADM network elements must support NETCONF monitoring [RFC 6022] including support for retrieving the NETCONF capabilities, datastores, schema and session information. Implementations may support NETCONF monitoring statistics.

### 3.3.4 Change notifications.

The Open ROADM YANG model supports the change-notification to identify changes on the network element. This notification provides information about the change including the date and time of the change, who initiated the change (server or user), the datastore affected, the change operation, and the target of the change. The target points to the element in the data model that has changed. It is up to the controller to retrieve that element to determine the new value.

It is recommended that in the event of a large number of changes, that the controller implement a holdoff and a consolidation in order to reduce the number of queries made to the device. MSA members may discuss future enhancements to include the changed data in the notification itself to avoid subsequent queries to the device.

### 3.3.5 YANG modeling

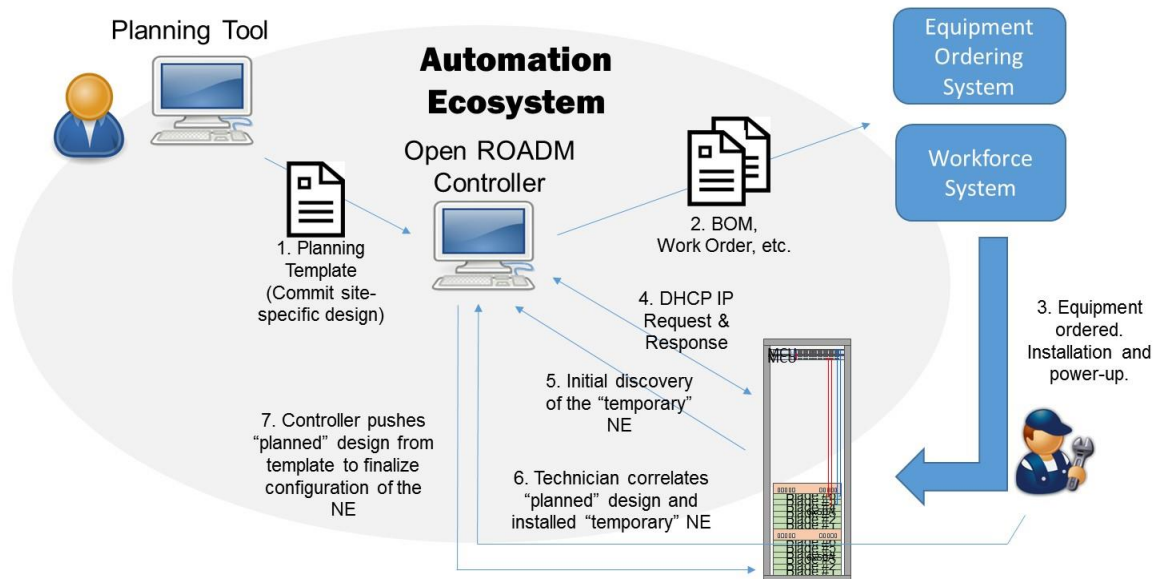
The Open ROADM data models are based on YANG v1 (RFC 6020). The support for YANG v1.1 [RFC 7950] is under consideration for future versions of Open ROADM.

## 3.4 OPEN ROADM DISCOVERY AND COMMISSIONING

The Open ROADM network element is provisioned through a one touch procedure to simplify the node commissioning step.



# One Touch for Network Automation

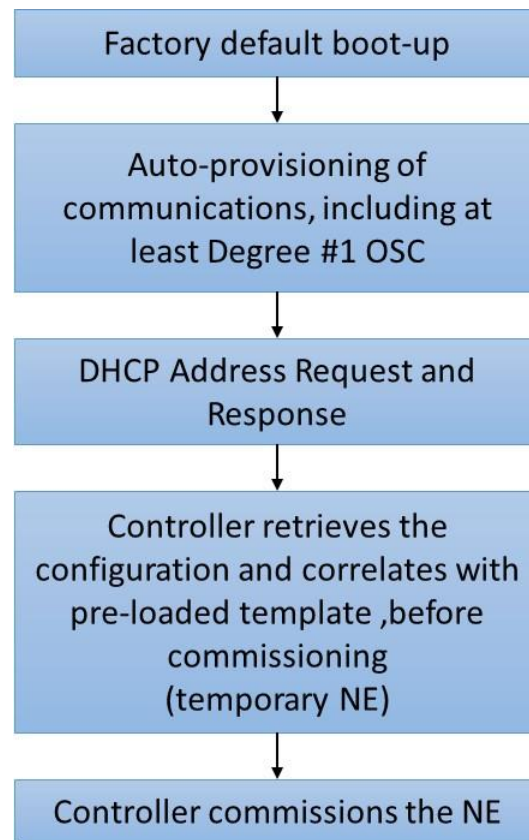


The steps for the commissioning include:

- Step 1: Loading the planning template into the Open ROADM controller (the specific method for loading the template is not standardized by Open ROADM). The planning template is not standardized by Open ROADM but provides data to the controller on how to commission the node using the Open ROADM device model. One possible implementation of the planning template is a JSON file containing a subset of the device model that would be needed to configure the Open ROADM device.
- Step 2: The Open ROADM controller may then generate information for ordering and work force systems to automate the installation.
- Step 3: The equipment is ordered. A field technician installs and powers the equipment.
- Step 4: The Open ROADM device initialization, autoprovisioning and IP address request via DHCP. The DHCP server responds with a temporary IP address.
- Step 5: The controller discovers the new IP address assignment by the DHCP server and attempts to connect and login to the device as an Open ROADM NE. If the device is an Open ROADM device, then the controller discovers the Open ROADM NE as a temporary NE.
- Step 6: The field technician provides the correlation between the controller discovered temporary NE and the pre-loaded planning template [One Touch]
- Step 7: The controller then pushes template configuration to the NE and rediscovers the NE (permanent NE)

Note: this section describes an example of a set of operations by an Open ROADM controller to commission a node. The MSA does not specify the operations required by an Open ROADM controller. Instead, it provides an example of how this could be done by a controller. It is possible that a controller could implement the commissioning steps differently than specified in this section.

The states on the network element are shown below:



#### 3.4.1 Pre-planned device template

The controller is loaded with the template for the node that is to be commissioned. The method for loading this template (from planning tool, manually configured, vendor provided, etc.) is not described in this document.

The template contains the information to provision the NE beyond the autoprovisioning behavior. This would include the final node-id, permanent IP address, shelf/circuit-pack/port attributes, etc.

#### 3.4.2 System Initialization and auto-provisioning

Upon system initialization, the Open ROADM NE will default the node-id to 'openroadm'. The default user account is 'openroadm' with password 'openroadm'.

Equipment should autoprovision to the point that communications can be established to the node, either through the external LAN port and/or the internal OSC. Both communication paths (external LAN and internal OSC) should be autoprovisioned as it will not be known which interface the NE will be accessed on.

Note: In general, the Open ROADM device should disable auto-provision with the exception of enabling the OSC for remote communications (minimum would be the OSC associated with degree #1). The reason to disable auto-provisioning is to prevent an accidental or incorrect auto-provisioning if the installer slots an equipment incorrectly. This would then create the entity associated with an incorrect

slot that would be difficult to remedy as it is assumed the installer will not have access to the management interface on the Open ROADM device.

For OSC connectivity, it should be sufficient to autoprovision the OSC for degree #1.

For equipment (circuit-packs) provisioned either automatically or manually, it is expected that the ports under the circuit-packs are auto-created by the NE and queryable via NETCONF.

### 3.4.3 DHCP

The Open ROADM network element initializes with DHCP client enabled. The NE requests both an IPv4 and IPv6 address as it is not known which protocol is in use in the carrier network. Preference is given to IPv6. This is the mechanism to obtain the initial IP address and connectivity to the NE. The method for querying both IPv4 and IPv6 (parallel/serial, interleaved, timing, etc) is device dependent.

Note: It is expected that the operator will later change the DHCP address to a permanent IP address during node commissioning so that the IP address will be stable. Please refer to Section 3.4.7.2 for further details.

### 3.4.4 Controller discovery of new IP address assignment

The controller discovers the DHCP IP address assignment from the DHCP server. The operator may choose to have the DHCP server coexists with their Open ROADM controller or have it located separately from the controller.

The mechanism for DHCP IP address assignment discovery is outside the scope of the Open ROADM specification.

### 3.4.5 Initial discovery of the network element

The controller cannot tell if a given DHCP IP address belongs to an Open ROADM network element, or some other IP capable device that supports DHCP (e.g., a laptop connected onto the DCN network). Thus, the controller must first try to connect to this address using NETCONF and see if the device supports the Open ROADM YANG models.

The controller will attempt to connect to the IP address with NETCONF over SSH using port 830. The SSH authentication uses 'openroadm' as the username and 'openroadm' as the password. If the login is successful, the controller will retrieve the info container to obtain the network element's vendor, model and serial number attributes.

At this time, the node is considered to be in a temporary state.

### 3.4.6 Correlation of the planned template and the temporary discovered node

The operator in the field connects to the controller to provide the correlation between a planned node in the controller's inventory and a newly discovered temporary node.

The operator would identify the planned node based on the site (CLLI) and node-id of the node he is installing.

The operator would identify the temporary discovered node based on the vendor, model and serial number of the equipment he installed.

Note: The operator needs to work with the equipment vendor about how to identify the serial number that will be used for correlation.

The operator would then correlate the planned node and discovered node to continue with the node commissioning.

### 3.4.7 Node commissioning.

Once the correlation has been made, the controller would begin to provision the network element. The template information is pushed to the network element using the NETCONF edit-config RPC with the merge operation. There may be processing involved in the controller, which takes both the template and the current state of the device as input, and determines the set of operations that need to be performed on the device.

The use of the merge operation allows the provisioning to succeed even if the entity (e.g., shelf, circuit-pack, port, etc.) was auto-provisioned due to the idempotent behavior.

The controller commissions the network element in the following order based on the Open ROADM device YANG models:

- Initial Info Container Validation
- Set Info container including setting the permanent node-id and IP address
- Shelf
- Circuit-pack, including ports
- Degree
- SRG
- Physical Links
- Interface (OAMP and OSC Ethernet, OTS, OMS)
- Protocol (LLDP and RSTP)
- User accounts
- Set date and time

Special handling is done for the user accounts and info container as described below.

#### 3.4.7.1 Initial Validation

The controller will validate the device's vendor and model information in the info container against the template setting to ensure we are provisioning the correct vendor device. This is done before configuring the device. If the vendor and model does not match, then an error is raised and the node commissioning stops.

#### 3.4.7.2 Node-ID, IP Address Provisioning

The first commissioning step is to update the info container, including setting the node-id, IP address (permanent IP address), prefix length and optional gateway.

The syntax for the node-id:

- length "7..20"
- pattern "([a-zA-Z][a-zA-Z0-9-]{5,18}[a-zA-Z0-9])"

When the IP address is changed, this will disable DHCP and remove the temporary IP address. Furthermore, by changing the IP address, the controller will re-establish the NETCONF session towards the permanent IP address.

After logging back into the NE, the device may require a restart using the following NETCONF RPC command:

```
<restart xmlns="http://org/openroadm/de/operations">
  <device>
    <node-id>openroadm</node-id>
  </device>
</resource/>
<resourceType>
  <type>device</type>
</resourceType>
<option>warm</option>
</restart>
```

Note that the node-id in the command above would be set to the actual node-id of the device that is being asked to be reset.

Following the reboot, the controller will relog back into the NE to continue node commissioning.

#### **3.4.7.3 User account provisioning**

The controller will provision the user accounts to be established on the network element, including the accounts (users) name, password and group. Currently only one group is defined called "sudo" that has full access to the NE.

The assumption under Open ROADM is that user security roles are managed at the controller, not on the NE.

The username is a string between 3-32 characters. The first character must be a lowercase letter. The remaining characters can be a lowercase letter or a number. The username may be treated as case insensitive on some devices.

The password is a string between 8-128 characters. Allowed characters in the password field include lowercase and uppercase letters, numbers and the special characters: ! \$ % ^ ( ) \_ + ~ { } [ ] . -

The following characters are not allowed in the password string: \ | ? # @ &

The default openroadm account will be deleted in a later step.

#### **3.4.7.4 Date and Time setting**

The controller will set the date and time directly on the NE. Note: the Open ROADM MSA does not require support for NTP or other time protocols in Open ROADM Version 1.2.1 (nor does the Open ROADM device model support provisioning of the address of time servers).

#### 3.4.7.5 *Delete default openroadm account*

The controller will log off the NE under the 'openroadm' account, and relog back into the NE using one of the new accounts. The controller will then delete the openroadm account from the NE and close the session.

The reason for this procedure is that you cannot delete an account while connected to the NE using that account.

#### 3.4.8 Permanent Node Discovery

After all commissioning steps are completed, the controller will discover the NE as a permanent NE under a new session.

At this point, the node is considered commissioned and fully discovered.

### 3.5 OMS LINK PLANNING AND DISCOVERY

OMS link data is pushed to the controller using the Open ROADM network model<sup>3</sup>. The OMS link plan would include the near end and far end degree information on the two sides of the link.

The Open ROADM device supports the LLDP protocol over the Ethernet OSC. LLDP is used to support the discovery of the link in the network.

Two key fields are used for the link discovery in the LLDP message:

- SysName (TLV Type 5): contains the NE's node-id
- PortID (TLV Type 2, sub-type 5 interface name or sub-type 7 locally assigned): contains the NE's OSC interface name

Open ROADM implementations may choose to send either the sub-type 5 or sub-type 7 PortID format. Whichever format is selected, the value that is sent should match the Ethernet OSC's interface name.

Open ROADM implementations should be able to process the receipt of either the sub-type 5 or sub-type 7 PortID.

Open ROADM device implementations should support the ability to receive and ignore other LLDP mandatory and optional TLVs that are not directly used by Open ROADM.

An LLDP change notification (lldp-nbr-info-change notification) is issued from the Open ROADM device whenever the neighbor LLDP information is changed. The controller uses this information to correlate the OSC transmit LLDP information with the OSC receive LLDP information to discover the link connectivity.

The discovered link is compared to the planned OMS link. If they match, then the controller will measure the span loss and take a baseline OTDR reading of the link to complete the OMS link commissioning.

---

<sup>3</sup> The Open ROADM network model resides on the Open ROADM controller (or other management system) and not on the device.

## 4 DEVICE SOFTWARE, FIRMWARE AND DATABASE

---

### 4.1 SOFTWARE UPGRADE

#### 4.1.1 Manifest file

See Section 9: Appendix A: Manifest file for software download and database operations

### 4.2 SOFTWARE DOWNGRADE

A software downgrade is not supported in-service in Open ROADM devices.

### 4.3 FIRMWARE UPGRADE

Firmware upgrade is not supported in Open ROADM Version 1.2.1. This feature will be added in Open ROADM Version 2.

### 4.4 DATABASE OPERATIONS

More detail on Database Operations will be added in the next version of this Whitepaper

#### 4.4.1 database backup

#### 4.4.2 database restore

#### 4.4.3 database change notifications

### 4.5 SYSLOG

More detail on Syslog will be added in the next version of this Whitepaper

## 5 NOTIFICATIONS

---

### 5.1 PM

The device model should include the following modules/models needed for network/service performance monitoring functions:

- A performance monitoring (PM) inventory list;
  - This is supported via the currentPmlist
    - Example invocation using NETCONF for retrieving current PM registers with a 15-minute granularity:

```
<get>
  <filter>
    <currentPmlist xmlns="http://org.openroadm/pm">
      <currentPm>
        <granularity>15min</granularity>
      </currentPm>
    </currentPmlist>
  </filter>
</get>
```

- Example invocation using NETCONF for retrieving current PM registers with a 24-hour granularity:

```
<get>
  <filter>
    <currentPmlist xmlns="http://org.openroadm/pm">
      <currentPm>
        <granularity>24Hour</granularity>
      </currentPm>
    </currentPmlist>
  </filter>
</get>
```



- A PM history collection that contains all the binned PM values collected by the device during some time window in 15-min intervals and 1-day intervals.
  - This is supported via the historyPmlist
    - Example invocation using NETCONF for retrieving bin 1 history PM registers with a 15-minute granularity:

```

<get>
  <filter>
    <historicalPmlist xmlns="http://org.openroadm/pm">
      <historicalPm>
        <id/>
        <resource/>
        <layerRate/>
        <binned-pm>
          <bin-number>1</bin-number>
        </binned-pm>
        <granularity>15min</granularity>
      </historicalPm>
    </historicalPmlist>
  </filter>
</get>

```

- Example invocation using NETCONF for retrieving bin 1 history PM registers with a 24-hour granularity:

```

<get>
  <filter>
    <historicalPmlist xmlns="http://org.openroadm/pm">
      <historicalPm>
        <id/>
        <resource/>
        <layerRate/>
        <binned-pm>

```

```

        <bin-number>1</bin-number>
    </binned-pm>
    <granularity>24Hour</granularity>
</historicalPm>
</historicalPmlist>
</filter>
</get>

```

- A configurable option to enable/disable PM event forwarding for any specific resource ids or a group of resource ids (under the resource node);
  - This functionality is not supported in Open ROADM Version 1.2.1.
- A configurable option to enable/disable PM event forwarding for any specific or combination of fault type (s) and time intervals (under pm node);
  - This functionality is not supported in Open ROADM Version 1.2.1.
- RPCs used to support configuration of PM event forwarding and PM event sync (after failed communication to northbound is restored or on-demand).
  - This functionality is not supported in Open ROADM Version 1.2.1.

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
Optical Power Output (OPOUT-OTS)	MW	OTS-IF	x		opticalPowerOutput		total optical signal power with OSC, includes VOA attenuation
				min	vendorExtension	opticalPowerOutputMin	
				max	vendorExtension	opticalPowerOutputMax	
				avg	vendorExtension	opticalPowerOutputAvg	

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
Optical Power Input (OPIN- OTS)	MW	OTS-IF	x		opticalPowerInput		total optical signal power with OSC
				min	vendorExtension	opticalPowerInputMin	
				max	vendorExtension	opticalPowerInputMax	
				avg	vendorExtension	opticalPowerInputAvg	
Optical Power Output (OPOUT-OMS)	MW	OMS-IF			opticalPowerOutput		total optical signal power without OSC, includes VOA attenuation
				min	vendorExtension	opticalPowerOutputMin	
				max	vendorExtension	opticalPowerOutputMax	
				avg	vendorExtension	opticalPowerOutputAvg	
Optical Power Input (OPIN- OMS)	MW	OMS-IF			opticalPowerInput		total optical signal power without OSC
				min	vendorExtension	opticalPowerInputMin	
				max	vendorExtension	opticalPowerInputMax	
				avg	vendorExtension	opticalPowerInputAvg	
Optical Return Loss (ORL-OTS)	MW	OTS-IF on tx port	x		opticalReturnLoss		at MW port(s) B (including OSC reflection) No tide markings required (e.g. min, max, avg) Either ORL-OTS or ORL-OMS is supported

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
							depending upon vendor implementation. direction, if included is rx (not tx)
Optical Return Loss (ORL-OMS)	MW	OMS-IF on tx port	x		opticalReturnLoss		at MW port(s) B (not including OSC reflection) No tide markings required (e.g. min, max, avg) Either ORL-OTS or ORL-OMS is supported depending upon vendor implementation. direction, if included is rx (not tx)
OSC Optical Power Transmit (OPT-OSC)	MW	OTS-IF			vendorExtension	opticalPowerOutputOSC	OSC Transmit power on MW port
				min	vendorExtension	opticalPowerOutputOSCMin	
				max	vendorExtension	opticalPowerOutputOSCMax	
				avg	vendorExtension	opticalPowerOutputOSCAvg	
OSC Optical Power Receive (OPR-OSC)	MW	OTS-IF			vendorExtension	opticalPowerInputOSC	OSC Receive power on MW port

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment		
					Type	extension			
						min		vendorExtension	opticalPowerInputOSCMIn
						max		vendorExtension	opticalPowerInputOSCMMax
		avg	vendorExtension	opticalPowerInputOSCAvg					
Optical Channel Power Transmit (OPT-OCH)	MW	OCH-IF	x		opticalPowerOutput		individual optical signal power on MW port		
				min	vendorExtension	opticalPowerOutputMin			
				max	vendorExtension	opticalPowerOutputMax			
				avg	vendorExtension	opticalPowerOutputAvg			
Optical Channel Power Receive (OPR-OCH)	MW	OCH-IF			opticalPowerInput		individual optical signal power on MW port, includes VOA attenuation		
				min	vendorExtension	opticalPowerInputMin			
				max	vendorExtension	opticalPowerInputMax			
				avg	vendorExtension	opticalPowerInputAvg			
Optical Power Receive (OPR)	Wr	SRG client port	x		opticalPowerInput		optical total signal power on Wr port (from transponder) – single wavelength		
				min	vendorExtension	opticalPowerInputMin			
				max	vendorExtension	opticalPowerInputMax			
				avg	vendorExtension	opticalPowerInputAvg			
Optical Power Transmit (OPT)	Wr	SRG client port			opticalPowerOutput		optical total signal power on Wr port (to		

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
							transponder) – multiple wavelengths
				min	vendorExtension	opticalPowerOutputMin	
				max	vendorExtension	opticalPowerOutputMax	
				avg	vendorExtension	opticalPowerOutputAvg	
Code Violations (CV-PCS)	OSC	Eth-IF			codeViolations		IEEE 802.3ah, Section 45.2.1.44; 8B/10B errors
Errored Seconds Section (ES-PCS)	OSC	Eth-IF			erroredSecondsSection		IEEE 802.3ah, Section 45.2.1.46
Severely Errored Seconds (SES-PCS)	OSC	Eth-IF			severlyErrorSeconds ( <i>typo</i> )		IEEE 802.3ah, Section 45.2.1.48 ( <i>typo will be fixed in next release</i> )
Unavailable Seconds PCS (UAS-PCS)	OSC	Eth-IF			unavailableSecondsPCS		IEEE 802.3ah, Section 45.2.1.52
In frames (INFRAMES-E)	OSC	Eth-IF			inFrames		
In frames errored (INFRAMESERR-E)	OSC	Eth-IF			inFramesErrored		
Out frames (OUTFRAMES-E)	OSC	Eth-IF			outFrames		

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
Errored Seconds Ethernet (ES-E)	OSC	Eth-IF			erroredSecondsEthernet		IEEE 802.3ah, Section 45.2.1.46
Severely Errored Seconds Ethernet (SES-E)	OSC	Eth-IF			severelyErroredSecondsEthernet		IEEE 802.3ah, Section 45.2.1.48
Unavailable Seconds Ethernet (UAS-E)	OSC	Eth-IF			unavailableSecondsEthernet		IEEE 802.3ah, Section 45.2.1.52
OSC Optical Power Receive (OPR-OSC)	OSC	port			opticalPowerInput		OSC Receive power on OSC port
				min	vendorExtension	opticalPowerInputMin	
				max	vendorExtension	opticalPowerInputMax	
				avg	vendorExtension	opticalPowerInputAvg	
OSC Optical Power Transmit (OPT-OSC)	OSC	port			opticalPowerOutput		OSC Transmit power on OSC port
				min	vendorExtension	opticalPowerOutputMin	
				max	vendorExtension	opticalPowerOutputMax	
				avg	vendorExtension	opticalPowerOutputAvg	
Current Output Power	W	port			opticalPowerOutput		Single wavelength to Wr
				min	vendorExtension	opticalPowerOutputMin	
				max	vendorExtension	opticalPowerOutputMax	
				avg	vendorExtension	opticalPowerOutputAvg	

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
Current Input Power	W	OCH-IF			opticalPowerInput		Single tuned wavelength
				min	vendorExtension	opticalPowerInputMin	
				max	vendorExtension	opticalPowerInputMax	
				avg	vendorExtension	opticalPowerInputAvg	
Total Input Power	W	port			totalOpticalPowerInput		Multiple wavelengths from Wr
				min	vendorExtension	totalOpticalPowerInputMin	
				max	vendorExtension	totalOpticalPowerInputMax	
				avg	vendorExtension	totalOpticalPowerInputAvg	
pFECcorrErr	W	OTU-IF on transponder network RX port	x		vendorExtension	preFECCorrectedErrors	Ref) G798 : 6.5.1.3
RX_ Forward Error Correction Correctable Blocks	W	OTU-IF			vendorExtension	FECCorrectableBlocks	count of corrected FEC Blocks direction = rx
RX_ Forward Error Correction Uncorrectable Blocks	W	OTU-IF			vendorExtension	FECUncorrectableBlocks	count of uncorrected FEC Blocks direction = rx
pN_EBC (BIP-8)	W	OTU-IF ODU-IF			erroredBlockCount		Ref) G.798 : 6.5.1.1 location = nearEnd
	W	ODU-IF			vendorExtension	erroredBlockCountTCM1	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM2	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM3	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM4	



MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM5	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM6	
	W	OTU-IF ODU-IF			erroredBlockCount		
pF_EBC (BEI)	W	ODU-IF			vendorExtension	erroredBlockCountTCM1	Ref) G.798 : 6.5.1.1 location = farEnd
	W	ODU-IF			vendorExtension	erroredBlockCountTCM2	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM3	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM4	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM5	
	W	ODU-IF			vendorExtension	erroredBlockCountTCM6	
	W	ODU-IF			delay		
pN_delay	W	ODU-IF			vendorExtension	delayTCM1	Ref) G798 : number of frames between a DMValue toggle event and the received DMP signal value toggle event location = nearEnd
	W	ODU-IF			vendorExtension	delayTCM1	
	W	ODU-IF			vendorExtension	delayTCM1	
	W	ODU-IF			vendorExtension	delayTCM1	
	W	ODU-IF			vendorExtension	delayTCM1	
	W	ODU-IF			vendorExtension	delayTCM1	
	W	ODU-IF			vendorExtension	delayTCM1	
RX_ Errored blocks	W	ETH-IF			erroredBlockCount		Ref) IEEE 802.3ba Section 45.2.3.12.4 direction = rx
RX_ BIP error counter	W	ETH-IF			vendorExtension	BIPErrorCounter	Ref) IEEE 802.3ba Section 45.2.3.36

MSA PM	MW Wr OSC	Reported Against...	Used by Controller	Tide Marks	Open ROADM yang version 1.2.1		Comment
					Type	extension	
							Ref) IEEE 802.3ba Section 45.2.3.37 direction = rx
TX_ Errored blocks	W	ETH-IF			erroredBlockCount		Ref) IEEE 802.3ba Section 45.2.3.12.4 direction = tx
TX_ BIP error counter	W	ETH-IF			vendorExtension	BIPErrorCounter	Sum of each lane is notified in FW9500. Ref) IEEE 802.3ba Section 45.2.3.36 Ref) IEEE 802.3ba Section 45.2.3.37 direction = tx

## 5.2 ALARMS

MSA Alarm	comment	cause	extension	resource resourceType	severity	direction *	location *
W Port Alarms							
OTU - LOS 100GE Signal Det	<p>OTN : It is shown that the input power drop. Ethernet : Ref) IEEE 802.3ba Section 86.2</p> <p>Physical maps to port level LOS</p> <p>Ethernet - linkDown at eth-if, LOS at sub-ports; LOS at parent port if all sub-ports LOS. (sub-port LOS is suppressed when port level LOS is raised. sub-port LOS is cleared and not retrievable when port level LOS is raised.)</p>	portLossOfLight		port			
	OCH LOS at transponder only (specific to tuned wavelength); Reported on both the network and client och; the network och alarm is used by the controller..)	lossOfSignal		och-if			

		linkDown		eth-if			
LOF	Ref) G798 : 6.2.5.1	lossOfFrame		otu-if			
LOM	Ref) G798 : 6.2.5.2	lossOfMultiframe		otu-if			
BDI	Ref) G798:6.2.6.6	backwardsDefectIndication		otu-if odu-if			
		backwardsDefectIndicationTCM1		odu-if			
		backwardsDefectIndicationTCM2		odu-if			
		backwardsDefectIndicationTCM3		odu-if			
		backwardsDefectIndicationTCM4		odu-if			
		backwardsDefectIndicationTCM5		odu-if			
		backwardsDefectIndicationTCM6		odu-if			
DEG	Ref) G798:6.2.3 ( <i>typo will be fixed in next release</i> )	degradedEffect ( <i>typo</i> )		otu-if odu-if			
		degradedDefectTCM1		odu-if			
		degradedDefectTCM2		odu-if			
		degradedDefectTCM3		odu-if			
		degradedDefectTCM4		odu-if			
		degradedDefectTCM5		odu-if			
		degradedDefectTCM6		odu-if			
BIAE	Ref) G798:6.2.6.11 * indicates non-alarmed standing condition in the MSA.  Non-alarmed is being	backwardIncomingAlignmentError		otu-if	warning		FEND
		backwardIncomingAlignmentErrorTCM1		odu-if	warning		FEND

	interpreted to mean "warning" severity.	backwardIncomingAlignmentErrorTCM2		odu-if	warning		FEND
		backwardIncomingAlignmentErrorTCM3		odu-if	warning		FEND
		backwardIncomingAlignmentErrorTCM4		odu-if	warning		FEND
		backwardIncomingAlignmentErrorTCM5		odu-if	warning		FEND
		backwardIncomingAlignmentErrorTCM6		odu-if	warning		FEND
IAE	Ref) G798:6.2.6.10 * indicates non-alarmed standing condition in the MSA.  Non-alarmed is being interpreted to mean "warning" severity.	incomingAlignmentError		otu-if	warning		FEND
		incomingAlignmentErrorTCM1		odu-if	warning		FEND
		incomingAlignmentErrorTCM2		odu-if	warning		FEND
		incomingAlignmentErrorTCM3		odu-if	warning		FEND
		incomingAlignmentErrorTCM4		odu-if	warning		FEND
		incomingAlignmentErrorTCM5		odu-if	warning		FEND
		incomingAlignmentErrorTCM6		odu-if	warning		FEND
TTI	Ref) G798:6.2.2.1	trailTraceIdentifierMismatch		otu-if odu-if			
		trailTraceIdentifierMisMatchTCM1		odu-if			
		trailTraceIdentifierMisMatchTCM2		odu-if			
		trailTraceIdentifierMisMatchTCM3		odu-if			
		trailTraceIdentifierMisMatchTCM4		odu-if			
		trailTraceIdentifierMisMatchTCM5		odu-if			
		trailTraceIdentifierMisMatchTCM6		odu-if			
AIS	Ref) G798:6.2.6.3.2	alarmIndicationSignal		otu-if odu-if			
		alarmIndicationSignalTCM1		odu-if			
		alarmIndicationSignalTCM2		odu-if			

		alarmIndicationSignalTCM3		odu-if			
		alarmIndicationSignalTCM4		odu-if			
		alarmIndicationSignalTCM5		odu-if			
		alarmIndicationSignalTCM6		odu-if			
OCI	Ref) G798:6.2.6.8	openConnectionIndication		odu-if			
		openConnectionIndicationTCM1		odu-if			
		openConnectionIndicationTCM2		odu-if			
		openConnectionIndicationTCM3		odu-if			
		openConnectionIndicationTCM4		odu-if			
		openConnectionIndicationTCM5		odu-if			
		openConnectionIndicationTCM6		odu-if			
LCK	Ref) G798:6.2.6.9	lockedDefect		odu-if			
		lockedDefectTCM1		odu-if			
		lockedDefectTCM2		odu-if			
		lockedDefectTCM3		odu-if			
		lockedDefectTCM4		odu-if			
		lockedDefectTCM5		odu-if			
		lockedDefectTCM6		odu-if			
LTC	Ref) G798:6.2.1.4	lossOfTandemConnectionTCM1		odu-if			
		lossOfTandemConnectionTCM2		odu-if			
		lossOfTandemConnectionTCM3		odu-if			
		lossOfTandemConnectionTCM4		odu-if			
		lossOfTandemConnectionTCM5		odu-if			
		lossOfTandemConnectionTCM6		odu-if			
PLM	Ref) G798 : 6.2.4.1	payloadMismatch		opu-if			
CSF	Ref) G798:6.2.10	clientSignalFailDefect		opu-if			
Loss of FEC Alignmen t	Only BjFEC=ON (i.e 100GBASE-SR4) Ref) IEEE 802.3bj Section 45.2.1.92b.2	lossOfFECAlignment		eth-if			

Loss of Sync (Rcv)	Ref) IEEE 802.3ba Section 45.2.3.11.5	lossOfSynchronization		eth-if		rx	
Rx_BASE-R PCS high BER	Ref) IEEE 802.3ba Section 45.2.3.11.4	highBER		eth-if		rx	
Loss of Alignment (Rcv)	OR of each lane is notified. Ref) IEEE 802.3ba Section 45.2.3.16c/d/e/f	lossOfAlignment		eth-if		rx	
Rx_Local Fault	Ref) IEEE 802.3ba Section 81.3.4	localFault		eth-if		rx	
Rx_Remote Fault	Ref) IEEE 802.3ba Section 81.3.4	remoteFault		eth-if		rx	
Loss of Sync (Tx)	Ref) IEEE 802.3ba Section 45.2.3.11.5	lossOfSynchronization		eth-if		tx	
Tx_BASE-R PCS high BER	Ref) IEEE 802.3ba Section 45.2.3.11.4	highBER		eth-if		tx	
Loss of Alignment (Tx)	OR of each lane is notified. Ref) IEEE 802.3ba Section 45.2.3.16c/d/e/f	lossOfAlignment		eth-if		tx	
Tx_Local Fault	Ref) IEEE 802.3ba Section 81.3.4	localFault		eth-if		tx	
Tx_Remote Fault	Ref) IEEE 802.3ba Section 81.3.4	remoteFault		eth-if		tx	
<b>MW Mux (Out) Ports</b>							
Automatic Shutoff/ Automatic Laser Shutdown	shut off amplifier laser due to safety, OSC will never be shut off	automaticLaserShutdown		oms-if			

Automatic Shutoff Disabled	RPC was triggered to shut off laser safety, standing alarm	automaticShutoffDisabled		oms-if			
Automatic Power Reduction Active (optional)	during high reflection event, power is lowered, but not shut off	vendorExtension	automaticPowerReduction	port			
High Reflection / Low Optical Return Loss	reflectionTooHigh on ots-if is used by the controller  Reported on ots-if or oms-if based on vendor implementation (ots-if if OSC is included in the measurement; oms-if, otherwise.	reflectionTooHigh		ots-if or oms-if			
Loss of Signal	May be used by the controller; Not masked since there is no LOS on OTS (tx); Optional for device to report	lossOfSignal		och-if		tx	
MW Demux (In) Ports							



Loss of Signal	LOS on OTS includes OSC and OMS;  LOS on OTS masks LOS on OMS but does not mask LOS on OSC and OMS  LOS on OTS is used by the controller.	lossOfSignal		ots-if		rx	
	LOS on OMS	lossOfSignal		oms-if		rx	
	LOS on OSC	vendorExtension	lossOfSignalOSC	ots-if		rx	
	LOS on OCH Masked by LOS on OTS (rx); Optional for device to report	lossOfSignal		och-if		rx	
Link Down	Link Down on OSC	linkDown		eth-if		rx	
Low Optical Power Warning (Optical Power Degraded )		opticalPowerDegraded		ots-if		rx	

Optical Line Fail	combines OMS LOS and linkDown on OSC into one alarm;  Optical Line Fail masks OMS LOS but does not mask linkDown on OSC and Ethernet linkdown	opticalLineFail		otsom s-if		rx	NEND
Wr Mux (In) Ports							
Loss of Signal		portLossOfLight		port		rx	
Optical Power Degraded		opticalPowerDegraded		port		rx	
The remaining causes do not map to any MSA Alarms currently							
		administrativeDown					
		certificateNotInstalled					
		databaseCorruption					
		databaseLockedDbAlarmPresent					
		databaseLockedDbRestoreInProgress					
	To be removed in version 2.0	databaseLockedIlfViolation					
		databaseLockedShelfProvModePresent					
		databaseLockedSoftwareUpgradeInProgress					

	To be removed in version 2.0	databaseLockedSysInitInProgress					
		databaseVersionMismatch					
		equipmentFault					
		equipmentInterConnectFailure					
		equipmentLedOn					
		equipmentMiscabledConnection					
		equipmentMismatch					
		equipmentRemoved					
		equipmentWarmup					
		facilityLoopback2Active					
		facilityLoopbackActive					
		facilityTestsignalActive					
		fanCoolingFail					
		firmwareBackwardCompatibleAll					
		firmwareBackwardCompatibleLimited					
		firmwareDownloadOrActivationFailure					
		firmwareInitInProgress					
		firmwareVersionMismatch					
		forwardDefectIndication					
	To be removed in version 2.0	ilfViolationCritical					
	To be removed in version 2.0	ilfViolationMajor					
		incompatibleFirmware					
		lampTest					
		oscPowerOutOfSpecificationHigh					
		oscPowerOutOfSpecificationLow					
		otsSpanlossPowerOutOfSpecificationHigh					

	To be removed in version 2.0	payloadMissingIndication					
	To be removed in version 2.0	postBlockSpanAdjustmentInProgress					
	To be removed in version 2.0	postBlockManualLaserShutdown					
		powerOutOfSpecificationHigh					
		powerProblemA					
		powerProblemB					
	To be removed in version 2.0	preBlockSpanAdjustmentInProgress					
		serverSignalFail					
		shelfProvisioningMode					
		softwareReset					
		softwareStageInProgress					
		softwareVersionMismatch					
		sysNameChanged					
		sysNtpNotSynchronized					
		terminalLoopbackActive					
		terminalTestsignalActive					

### 5.3 THRESHOLD CROSSING ALARMS (TCAs)

TCAs have some bugs in Open ROADM Version 1.2.1 (e.g. only integer values). The plan is to address these issues in Open ROADM V2 (some initial fixes are in Version 2.0).

### 5.4 OPERATIONAL STATES AND DATABASE CHANGE NOTIFICATIONS

More detail on this will be added in the next version of this Whitepaper

## 6 DEVICE SPECIFIC TOPICS

---

### 6.1 ROADM

#### 6.1.1 OTDR scan

#### 6.1.2 Interfaces

also need to talk about GitHub issue 108, MPO modelling

<< Need to add the hierarchy for interfaces on the ROADM

### 6.2 TRANSPONDER

#### 6.2.1 Interfaces

<< Need to add the hierarchy for interfaces on the ROADM

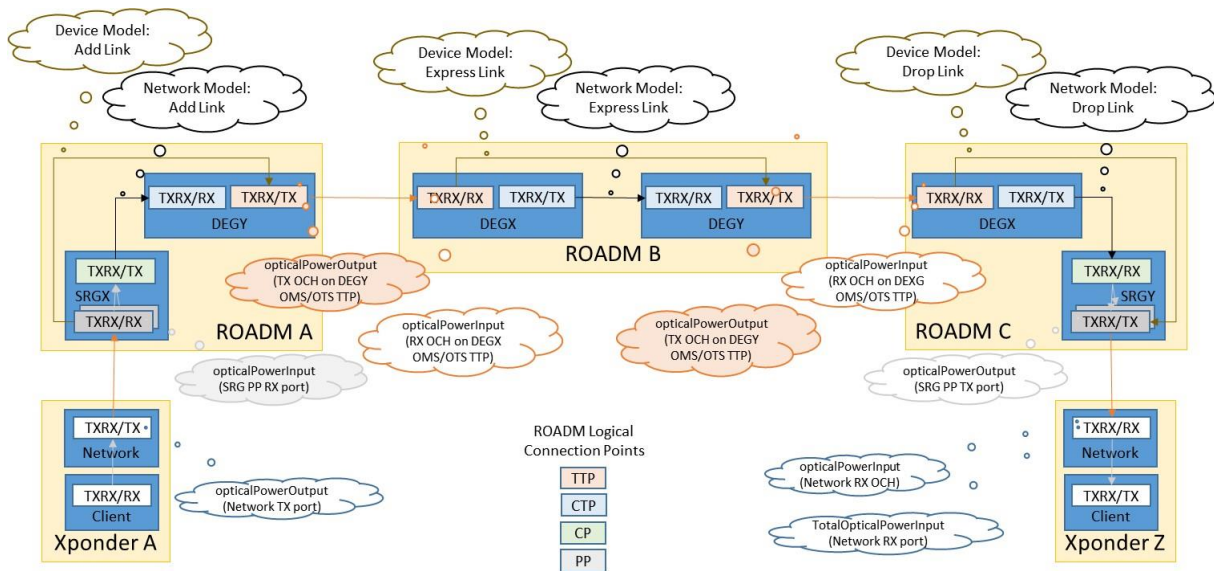
### 6.3 PLUGGABLE

In V1.2.1, no pluggable stand-alone device (e.g. WDM line side pluggable directly in a router) exists yet in the device model. This will be tackled in V2 and beyond.

## 7 PROVISIONING ACTION USE CASES

---

The following figure is used to provide context for this section.



## 7.1 CONNECTION MANAGEMENT

In this section, the different basic operations needed to configure interfaces and connections on the ROADM transponder nodes are presented. Interface and connections deletion is presented in a second step. These actions are performed through device's NETCONF interfaces. XML encoding is not detailed and only basic NETCONF operations are presented in order to simplify and alleviate the description.

In the same way, because some ports/interfaces can be either uni (TX/RX) or bi-directional (TXRX), we try to be as generic as possible in the action sequence description. Consequently, such ports will be specified as [TXRX/TX] or [TXRX/RX].

### 7.1.1 Xponder interfaces creation (TX)

Configures the ports and interfaces on both line and client sides of a transponder in the TX direction. Xponder interfaces are supposed to be bi-directional. However, we describe the sequence of actions following the provisioning methodology; where the connections are configured sequentially across the pre-calculated path in one direction first, and then in the reverse direction. Thus, we only give precision on the elements and configuration actions that will be engaged at a time for a specific purpose. Real implementation may differ from the proposed sequence of action. As another example, we consider applying power settings just after the connection is created, but they could be performed in a second step.

#### **Input parameters:**

- node-id (xponder X)
- xponder network circuit-pack-name (xponder X network circuit pack)
- xponder network port-name (xponder X network port)
- xponder client circuit-pack-name (xponder X client circuit pack)
- xponder client port-name (xponder X client port)
- wavelength-number

- rate
- modulation-format
- transmit-power

**Main actions sequence:**

Get-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]/ports[port-name= "<xponder X network port>"])/retrieve xponder X network port transponder-port/capable-wavelengths

retrieve xponder X network port transponder-port/port-power-capability (min/max tx)

retrieve xponder X network port supported-interface-capability

Get-config (org-openroadm-device/interface[supporting-circuit-pack-name="<xponder X network circuit pack >" and supporting-port="<xponder X network port >"])

Check if OCH interface present on network port and the status of the port

**If NO OCH corresponding interface created on the Xponder network port**

*(either bi-directional port or uni-directional TX port)*

equipment-state of circuit-pack and admin-state of network-port shall be set appropriately :

edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]) nc:operation=replace

set equipment-state of OCH interface supporting circuit-pack to not-reserved-inuse

edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]/ports[port-name= "<xponder X network port>"]) nc:operation=replace

set admin state of OCH interface supporting port to inService

OCH interface shall be created on network port:

edit-config (org-openroadm-device/interface>") nc:operation=create

create an OCH interface XPDR<n>-NETWORK<m>-[TXRX/TX]-[λ-number] on supporting port

set name = "<XPDR<n>-NETWORK<m>-[TXRX/TX]-[λ-number] >"

set type to opticalChannel

set supporting-circuit-pack-name="<xponder X network circuit pack >"

set supporting-port ="<xponder X network port >"

set rate, wavelength-number, modulation-format

set admin state of OCH interface to inService

OTU4 interface shall be created on supporting OCH interface:

edit-config (org-openroadm-device/interface) nc:operation=create

create an otnOtu interface OTU4 on supporting port/OCH interface

set name = "<OTU4-XPDR<n>-NETWORK<m>-TXRX/TX>"

set type to otnOtu

set supporting-circuit-pack-name="<xponder X network circuit pack >"

set supporting-port ="<xponder X network port >"

set supporting-interfaces ="< OCH interface>"

set admin state of OTU4 interface to inService

ODU4 interface shall be created on supporting OTU4 interface:

```
edit-config (org-openroadm-device/interface) nc:operation=create
```

create an otnOdu interface *ODU4* on supporting port/OTU4 interface

```
set name = "<ODU4-XPDR<n>-NETWORK<m>-TXRX/TX>"
set type to otnOdu
set supporting-circuit-pack-name="<xponder X network circuit pack >"
set supporting-port ="<xponder X network port >"
set supporting-interfaces "<OTU4-XPDR<n>-NETWORK<m>-TXRX/TX>"
set admin state of ODU4-1 interface to inService
```

Output power shall be configured on the OCH interface:

```
edit-config (org-openroadm-device/interface[name= "<XPDR<n>-NETWORK<m>-[TXRX/TX]-[λ-number]>"]/och/transmit-power) nc:operation=replace
```

set transmit-power to output-power

The controller verifies that the output power reaches the set value (verify pmParameterName/type = opticalPowerOutput):

```
Get-config (/currentPmList/currentPm/currentPmList/currentPm[resource/resource/port-name"<xponder X network port >"and granularity="15min"]])
```

The controller verifies that connection between client and network xponder circuit packs and ports exists:

Optional: 

```
Get-config (org-openroadm-device/connection-map[source/circuit-pack-name="<xponder X network circuit pack >"and source/port-name="<xponder X client port>" and destination/circuit-pack-name="<xponder X client circuit pack > and destination/port-name="<xponder X network port>"]])
```

Check if interface present on client port and the status of the port

```
Get-config (org-openroadm-device/interface[supporting-circuit-pack-name="<xponder X client circuit pack >" and supporting-port-name="<xponder X client port >"]])
```

***If NO corresponding interface created on the Xponder client port***

*(either bidirectional port or uni-directional RX port)*

equipment-state of circuit-pack and admin-state of network-port shall be set appropriately :

```
edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X client circuit pack>"]) nc:operation=replace
```

set equipment-state of 100GE interface supporting circuit-pack to not-reserved-inuse

```
edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X client circuit pack>"])/ports[port-name= "<xponder X client port>"]) nc:operation=replace
```

set admin state of 100GE interface supporting port to inService

100GE Ethernet interface shall be created on client port:

```
edit-config (org-openroadm-device/interface) nc:operation=create
```

create a 100GE interface 100GE on supporting port

```
set name = "<100GE-XPDR<n>-CLIENT<m>-TXRX/TX>"
```

```
set type to ethernetCsmacd
```



```

set supporting-circuit-pack-name="< xponder X client circuit pack >"
set supporting-port ="< xponder X client port >"
set speed, fec, duplex, mtu, auto-negotiation, curr-speed
set admin state of 100GE interface to inService

```

## END of sequence

### 7.1.2 Add-Link creation

Configures the ports, interfaces and connections for an added link from a SRG-PP to Degree TTP logical point. One shall note that the power settings could be applied in a second step, after the interfaces and the connection have been created. Some actions are optional and may differ from one implementation to another.

#### **Input parameters:**

- node-id (SRGX, DEGY)
- circuit-pack (SRGX CircuitPack, DEGY CircuitPack)
- supporting-port (SRGX-PPN, DEGY-TTP)
- wavelength number
- calculated output power

#### **Main actions sequence:**

When optional is mentioned, it usually refers to a step where the controller performs some verification steps that are not mandatory.

**Optional :** `Get-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<SRGX CircuitPack >"]/ports/roadm-port)`

retrieve SRG roadm-port available-wavelengths

retrieve SRG roadm-port used-wavelengths

**Optional :** `Get-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<DEGY CircuitPack >"]/ports/roadm-port)`

retrieve Degree roadm-port available-wavelengths

retrieve Degree roadm-port used-wavelengths

`Get-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<SRGX CircuitPack >"]/ports [port-name= "<SRGX-PPN >"]/interfaces)`

#### **If NO OCH (bidirectional) corresponding interface created on the SRGX-PPN**

`edit-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<SRGX CircuitPack >"]/ports [port-name= "<SRGX-PPN >"]/) nc:operation=replace`

set admin state of OCH interface supporting PP port to inService

`edit-config (org-openroadm-device/interface) nc:operation=create`

create OCH-interface 1 (SRGX-PPN-[TXRX/RX]-[λnumber]) on supporting PP port

set name = "<SRGX-PPN-[TXRX/RX]-[ λ-number] >"

set type to opticalChannel

```
set supporting-circuit-pack-name= "< SRGX CircuitPack >"
set supporting-port = "< SRGX-PPN >"
set wavelength-number
set admin state of OCH interface inService
```

```
Get-config (org-openroadm-device/ circuit-packs[circuit-pack-name= "<DEGY-CircuitPack>"]/ports[port-
name= "<DEGY-TTP>/interfaces")
```

***If NO OTS (bidirectional) corresponding interface created on the DEGY-TTP***

```
edit-config (org-openroadm-device/interface) nc:operation=create
create OTS-interface (OTS-DEGY-TTP-[TXRX/TX]) on supporting TTP port
set name = "<OTS-DEGY-TTP-[TXRX/TX]>"
set type to opticalTransport
set supporting-circuit-pack-name= "< DEGY CircuitPack >"
set supporting-port = "< DEGY-TTP >")
set admin state of OTS-interface to inService
```

***If NO OMS (bidirectional) corresponding interface created on the DEGY-TTP***

```
edit-config (org-openroadm-device/interface) nc:operation=create
create OMS-interface (OMS-DEGY-TTP-[TXRX/TX]) on supporting TTP port
set name = "<OMS-DEGY-TTP-[TXRX/TX]>"
set type to openROADMOpticalMuiltiplex
set supporting-circuit-pack-name= "< DEGY CircuitPack >"
set supporting-port = "< DEGY-TTP >")
set supporting-interfaces ="<OTS-DEGY-TTP-[TXRX/TX]>"
set admin state of OMS-interface to inService
```

***If NO OCH (bidirectional) corresponding interface created on the DEGY-TTP***

```
edit-config (org-openroadm-device/interface) nc:operation=create
create OCH-interface2 (DEGY-TTP-[TXRX/TX]-[λnumber]) on supporting TTP port
set name = "<DEGY-TTP-[TXRX/TX]-[λnumber]>"
set type to opticalChannel
set supporting-circuit-pack-name= "< DEGY CircuitPack >"
set supporting-port = "< DEGY-TTP >")
set supporting-interfaces ="<OMS-DEGY-TTP-[TXRX/TX]>"
set wavelength-number
set admin state of OCH-interface2 to inService
```

```
edit-config (org-openroadm-device/roadm-connections) nc:operation=create
```

```
Set connection from OCH -Interface1 to OCH-Interface 2
```

```
Set connection-number = "<SRGX-PPN-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-
[λnumber]>"
set wavelength-number
set source/src-if = "<SRGX-PPN-[TXRX/RX]-[ λ-number] >"
```

set destination/dst-if = "<DEGY-TTP-[TXRX/TX]-[λnumber]>"

*Power setting and control-mode change could be made in a second step after all connections are created*

`edit-config (org-openroadm-device/roadm-connections[connection-number= "<SRGX-PPN-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=replace`

set optical control-mode = power

set target-output-power = calculated output power

In a final step, the controller verifies that the output power reaches the set value (verify pmParameterName/type = opticalPowerOutput):

`Get-config (/currentPmList/currentPm[resource/resource/interface-name="< DEGY-TTP-[TXRX/TX]-[λnumber]>" and granularity="15min"])`

And once current-output-power has converged towards expected output power, control-mode is moved to gainLoss

`edit-config (org-openroadm-device/roadm-connections[connection-number= "<SRGX-PPN-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=replace`

Connection from OCH-Interface 1(PP) to OCH-Interface2 (TTP)

set optical control-mode = gainLoss

#### **End of sequence**

#### **7.1.3 Express link creation**

Configures the ports, interfaces and connections for an express link between 2 Degree TTP logical points in one direction. One shall note that the power settings could be applied in a second step, after the interfaces and the connections have been created. Some actions are optional and may differ from one implementation to another.

A bidirectional express link will be created in 2 steps (connections are unidirectional) applying this sequence in both directions.

#### **Input parameters:**

- node-id (DEGX, DEGY)
- circuit-pack (DEGX CircuitPack
- DEGY CircuitPack)
- supporting-port (DEGX-TTP, DEGY-TTP)
- wavelength number
- expected input power
- calculated output power

#### **Main actions sequence:**

Optional : `Get-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<DEGX CircuitPack >"]/ports/roadm-port)`

retrieve Degree roadm-port available-wavelengths

retrieve Degree roadm-port used-wavelengths

Optional : Get-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<DEGY CircuitPack >"]/ports/roadm-port)

retrieve Degree roadm-port available-wavelengths

retrieve Degree roadm-port used-wavelengths

Get-config (org-openroadm-device/ circuit-packs[circuit-pack-name= "<DEGX-CircuitPack>"]/ports[port-name= "<DEGX-TTP>"]/interfaces)

***If NO OTS (bidirectional) corresponding interface created on the DEGX-TTP***

edit-config (org-openroadm-device/interface) nc:operation=create

create OTS-interface (OTS-DEGX-TTP-[TXRX/RX]) on supporting TTP port

set name = "<OTS-DEGX-TTP-[TXRX/RX]>"

set type to opticalTransport

set supporting-circuit-pack-name= "< DEGX CircuitPack >"

set supporting-port = "< DEGX-TTP >")

set admin state of OTS-interface to inService

***If NO OMS (bidirectional) corresponding interface created on the DEGX-TTP***

edit-config (org-openroadm-device/interface) nc:operation=create

create OMS-interface (OMS-DEGX-TTP-[TXRX/RX]) on supporting TTP port

set name = "<OMS-DEGX-TTP-[TXRX/RX]>"

set type to openROADMOpticalMulitplex

set supporting-circuit-pack-name= "< DEGX CircuitPack >"

set supporting-port = "< DEGX-TTP >")

set supporting-interfaces ="<OTS-DEGX-TTP-[TXRX/RX]>"

set admin state of OMS-interface to inService

***If NO OCH (bidirectional) corresponding interface created on the DEGX-TTP***

edit-config (org-openroadm-device/interface) nc:operation=create

create OCH-interface1 (DEGX-TTP-[TXRX/RX]-[λnumber]) on supporting TTP port

set name = "<DEGX-TTP-[TXRX/RX]-[λnumber]>"

set type to opticalChannel

set supporting-circuit-pack-name= "< DEGX CircuitPack >"

set supporting-port = "< DEGX-TTP >")

set supporting-interfaces ="<OMS-DEGX-TTP-[TXRX/RX]>"

set wavelength-number

set admin state of OCH-interface1 to inService

The following step may be optional and aims at verifying that input power is compliant with MSA specifications (verify pmParameterName/type = opticalPowerInput):

Optional : Get-config (/currentPmlist/currentPm[resource/resource/interface-name="<DEGX-TTP-[TXRX/TX]-[λnumber]>" and granularity="15min"])

retrieve opticalPowerInput

compare to expected input power derived from MSA specifications : per channel output power of preceding span (per channel output power offset diagram)- span loss

`Get-config (org-openroadm-device/ circuit-packs[circuit-pack-name= "<DEGY-CircuitPack>"])/ports [port-name= "<DEGY-TTP>]/interfaces")`

***If NO OTS (bidirectional) corresponding interface created on the DEGY-TTP***

`edit-config (org-openroadm-device/interface) nc:operation=create`

create OTS-interface (OTS-DEGY-TTP-[TXRX/TX]) on supporting TTP port  
set name = "<OTS-DEGY-TTP-[TXRX/TX]>"  
set type to opticalTransport  
set supporting-circuit-pack-name= "< DEGY CircuitPack >"  
set supporting-port = "< DEGY-TTP >"  
set admin state of OTS-interface to inService

***If NO OMS (bidirectional) corresponding interface created on the DEGY-TTP***

`edit-config (org-openroadm-device/interface) nc:operation=create`

create OMS-interface (OMS-DEGY-TTP-[TXRX/TX]) on supporting TTP port  
set name = "<OMS-DEGY-TTP-[TXRX/TX]>"  
set type to openROADMOpticalMux  
set supporting-circuit-pack-name= "< DEGY CircuitPack >"  
set supporting-port = "< DEGY-TTP >"  
set supporting-interfaces ="<OTS-DEGY-TTP-[TXRX/TX]>"  
set admin state of OMS-interface to inService

***If NO OCH (bidirectional) corresponding interface created on the DEGY-TTP***

`edit-config (org-openroadm-device/interface) nc:operation=create`

create OCH-interface2 (DEGY-TTP-[TXRX/TX]-[λnumber]) on supporting TTP port  
set name = "<DEGY-TTP-[TXRX/TX]-[λnumber]>"  
set type to opticalChannel  
set supporting-circuit-pack-name= "< DEGY CircuitPack >"  
set supporting-port = "< DEGY-TTP >"  
set supporting-interfaces ="<OMS-DEGY-TTP-[TXRX/TX]>"  
set wavelength-number  
set admin state of OCH-interface2 to inService

`edit-config (org-openroadm-device/roadm-connections) nc:operation=create`

set connection from OCH -Interface1 to OCH-Interface 2  
set connection-number = "<DEGX-TTP-[TXRX/RX]-[λnumber]-to-DEGY-TTP-[TXRX/TX]-[λnumber]>"  
set wavelength-number  
set source/src-if = "<DEGX-TTP-[TXRX/TX]-[λnumber] >"  
set destination/dst-if = "<DEGY-TTP-[TXRX/TX]-[λnumber]>"

*Power setting and control-mode change could be made in a second step after all connections are created*

```
edit-config (org-openroadm-device/roadm-connections[connection-number= "<DEGX-TTP-[TXRX/RX]-[λnumber]-to-DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=replace
```

```
set optical control-mode = power
```

```
set target-output-power = calculated output power
```

The output power is calculated from MSA specifications: per channel output power offset diagram)

In a final step, the controller verifies that the output power reaches the set value (verify pmParameterName/type = opticalPowerOutput):

```
Get-config (/currentPmList/currentPm[resource/resource/interface-name="< DEGY-TTP-[TXRX/TX]-[λnumber]>" and granularity="15min"])
```

And once current-output-power has converged towards expected output power, control-mode is moved to gainLoss

```
edit-config (org-openroadm-device/roadm-connections[connection-number= "<DEGX-TTP-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=replace
```

Change optical-control-mode of connection from OCH-Interface 1(TTP) to OCH-Interface2 (TTP)

```
set optical control-mode = gainLoss
```

## End of Sequence

### 7.1.4 Drop-Link creation

Configures the ports, interfaces and connections for drop link between a TTP and a PP logical points.

One shall note that the power settings could be applied in a second step, after the interfaces and the connections have been created. Some actions are optional and may differ from one implementation to another.

#### **Input parameters:**

- node-id (DEGX, SRGY)
- circuit-pack (DEGX CircuitPack, SRGY CircuitPack)
- supporting-port (DEGX-TTP, SRGY-PPM)
- wavelength number
- calculated output power
- expected input power

#### **Main actions sequence:**

Optional : Get-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<DEGX CircuitPack >"]/ports/roadm-port)

```
retrieve Degree roadm-port available-wavelengths
```

```
retrieve Degree roadm-port used-wavelengths
```

Optional : Get-config(org-openroadm-device/circuit-packs[circuit-pack-name= "<SRGY CircuitPack >"]/ports/roadm-port)

```
retrieve SRG roadm-port available-wavelengths
```

```
retrieve SRG roadm-port used-wavelengths
```

Get-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<DEGX-CircuitPack>"]/ports[port-name= "<DEGX-TTP>"]/interfaces)

**If NO OTS (bidirectional) corresponding interface created on the DEGX-TTP**

edit-config (org-openroadm-device/interface) nc:operation=create

create OTS-interface (OTS-DEGX-TTP-[TXRX/RX]) on supporting TTP port  
set name = "<OTS-DEGX-TTP-[TXRX/RX]>"  
set type to opticalTransport  
set supporting-circuit-pack-name= "< DEGX CircuitPack >"  
set supporting-port = "< DEGX-TTP >")  
set admin state of OTS-interface to inService

**If NO OMS (bidirectional) corresponding interface created on the DEGX-TTP**

edit-config (org-openroadm-device/interface) nc:operation=create

create OMS-interface (OMS-DEGX-TTP-[TXRX/RX]) on supporting TTP port  
set name = "<OMS-DEGX-TTP-[TXRX/RX]>"  
set type to openROADMOpticalMuiltiplex  
set supporting-circuit-pack-name= "< DEGX CircuitPack >"  
set supporting-port = "< DEGX-TTP >")  
set supporting-interfaces ="<OTS-DEGX-TTP-[TXRX/RX]>"  
set admin state of OMS-interface to inService

**If NO OCH (bidirectional) corresponding interface created on the DEGX-TTP**

edit-config (org-openroadm-device/interface) nc:operation=create

create OCH-interface1 (DEGX-TTP-[TXRX/RX]-[λnumber]) on supporting TTP port  
set name = "<DEGX-TTP-[TXRX/RX]-[λnumber]>"  
set type to opticalChannel  
set supporting-circuit-pack-name= "< DEGX CircuitPack >"  
set supporting-port = "< DEGX-TTP >")  
set supporting-interfaces ="<OMS-DEGX-TTP-[TXRX/RX]>"  
set wavelength-number  
set admin state of OCH-interface1 to inService

The following step may be optional and aims at verifying that input power is compliant with MSA specifications (verify pmParameterName/type = opticalPowerInput):

Optional: Get-config (/currentPmList/currentPm[resource/resource/interface-name="<DEGX-TTP-[TXRX/RX]-[λnumber]>" and granularity="15min"])

retrieve opticalPowerInput

compare to expected input power derived from MSA specifications : per channel output power of preceding span (per channel output power offset diagram)- span loss

Get-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<SRGY CircuitPack >"]/ports[port-name= "<SRGY-PPN >"]/interfaces)

**If NO OCH (bidirectional) corresponding interface created on the SRGY-PPN**

```
edit-config(org-openroadm-device/circuit-packs/[circuit-pack-name= "<SRGY CircuitPack >"]/ports  
/[port-name= "<SRGY-PPN >"]/) nc:operation=replace
```

set admin state of OCH interface supporting PP port to inService

```
edit-config (org-openroadm-device/interface) nc:operation=create
```

create OCH-interface 2 (SRGY-PPN-[TXRX/TX]-[λnumber]) on supporting PP port

set name = "<SRGY-PPN-[TXRX/TX]-[ λ-number] >"

set type to opticalChannel

set supporting-circuit-pack-name= "< SRGY CircuitPack >"

set supporting-port = "< SRGY-PPN >"

set wavelength-number

set admin state of OCH interface 2 to inService

```
edit-config (org-openroadm-device/roadm-connections) nc:operation=create
```

Set connection from OCH-interface1 to OCH-interface2 DEGX-TTP-[TXRX/RX]-[λnumber]-to- SRGY-PPM-[TXRX/TX]-[λnumber]

Set connection-number = "<DEGX-TTP-[TXRX/RX]-[λnumber]-to- SRGY-PPM-[TXRX/TX]-[λnumber]>"

set wavelength-number

set source/src-if = "<DEGX-TTP-[TXRX/RX]-[λnumber]>"

set destination/dst-if = "<SRGY-PPN-[TXRX/TX]-[ λ-number] >"

set optical control-mode = power

In a final step, the controller verifies that the output power reaches the set value (verify pmParameterName/type = opticalPowerOutput):

```
Get-config (/currentPmList/currentPm[resource/resource/port-name="<SRGY-PPN>" and  
granularity="15min"])
```

retrieve SRG opticalPowerOutput

**End of sequence**

#### 7.1.5 Xponder interfaces creation (RX)

Configures the ports and interfaces on both line and client sides of a transponder in the RX direction.

**Input parameters:**

- node-id (xponder X)
- xponder network circuit-pack-name (xponder X network circuit pack)
- xponder network port-name (xponder X network port)
- xponder client circuit-pack-name (xponder X client circuit pack)
- xponder client port-name (xponder X client port)
- SRGY supporting port (SRGY-PPN)



- rate
- modulation format

**Main actions sequence:**

Get-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]/ports[port-name= "<xponder X network port>"]/transponder-port)

retrieve xponder X network port **port-power-capability** (min/max rx)

retrieve xponder X network port **supported-interface-capability**

Optional: Check SRGY-PPN opticalPowerOutput retrieved in drop-link creation is compatible with min-max rx-port-power-capability

Get-config (org-openroadm-device/interface[supporting-circuit-pack-name="<xponder X network circuit pack>" and supporting-port-name="<xponder X network port>"])

Check if OCH interface present on network port and the status of the port

**If NO OCH corresponding interface created on the Xponder network port**

*(either bi-directional port or uni-directional RX port)*

equipment-state of circuit-pack and admin-state of network-port shall be set appropriately :

edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]) nc:operation=replace

set equipment-state of OCH interface supporting circuit-pack to not-reserved-inuse

edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]/ports[port-name= "<xponder X network port>"]) nc:operation=replace

set admin state of OCH interface supporting port to inService

OCH interface shall be created on network port:

edit-config (org-openroadm-device/interface) nc:operation=create

create an OCH interface XPDR<n>-NETWORK<m>-[TXRX/RX]-[λ-number] on supporting port

set name = "<XPDR<n>-NETWORK<m>-[TXRX/RX]-[ λ-number] >"

set type to opticalChannel

set supporting-circuit-pack-name="<xponder X network circuit pack >"

set supporting-port ="<xponder X network port >"

set rate, wavelength-number, modulation-format

set admin state of OCH interface to inService"

OTU4 interface shall be created on supporting OCH interface:

edit-config (org-openroadm-device/interface) nc:operation=create

create an otnOtu interface OTU4 on supporting port/OCH interface

set name = "<OTU4-XPDR<n>-NETWORK<m>-TXRX/RX>"

set type to otnOtu

set supporting-circuit-pack-name="<xponder X network circuit pack >"

set supporting-port ="<xponder X network port >"

set supporting-interfaces ="< OCH interface>"

set admin state of OTU4 interface to inService  
 ODU4 interface shall be created on supporting OTU4 interface:  
 edit-config (org-openroadm-device/interface) nc:operation=create  
 create an otnOdu interface *ODU4* on supporting port/OTU4 interface  
 set name = "<ODU4-XPDR<n>-NETWORK<m>-TXRX/RX>"  
 set type to otnOdu  
 set supporting-circuit-pack-name="<xponder X network circuit pack >"  
 set supporting-port ="<xponder X network port >"  
 set supporting-interfaces ="<OTU4-XPDR<n>-NETWORK<m>-TXRX/RX>"  
 set admin state of ODU4 interface to inService

The controller verifies that connection between client and network xponder circuit packs exists:

Optional: Get-config (org-openroadm-device/connection-map[source/circuit-pack-name="<xponder X network circuit pack >" and source/port-name="<xponder X network port>" and destination/circuit-pack-name="<xponder X client circuit pack > and destination/port-name="<xponder X client port>"])

Check if interface present on client port and the status of the port

Get-config (org-openroadm-device/interface[supporting-circuit-pack-name="<xponder X client circuit pack >" and supporting-port-name="<xponder X client port >"])

***If NO corresponding interface created on the Xponder client port***

*(either bidirectional port or uni-directional TX port)*

equipment-state of circuit-pack and admin-state of network-port shall be set appropriately :

edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X client circuit pack>"]) nc:operation=replace

set equipment-state of 100GE interface supporting circuit-pack to not-reserved-inuse

edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X client circuit pack>"]/ports[port-name= "<xponder X client port>"]) nc:operation=replace

set admin state of 100GE interface supporting port to inService

100GE Ethernet interface shall be created on client port:

edit-config (org-openroadm-device/interface) nc:operation=create

create a 100GE interface 100GE on supporting port

set name = "<100GE-XPDR<n>-CLIENT<m>-TXRX/TX>"

set type to ethernetCsmacd

set supporting-circuit-pack-name="<xponder X client circuit pack >"

set supporting-port ="<xponder X client port >"

set speed, fec, duplex, mtu, auto-negotiation, curr-speed

set admin state of 100GE interface to inService

**End of sequence**

### 7.1.6 Xponder interfaces deletion (TX/RX)

Deletes the interfaces and un-configures the ports on both line and client sides of a transponder in both RX and TX directions.

#### **Input parameters:**

- node-id (xponder X)
- xponder network circuit-pack-name (xponder X network circuit pack)
- xponder network port-name (xponder X network port)
- xponder client circuit-pack-name (xponder X client circuit pack)
- xponder client port-name (xponder X client port)

#### **Main actions sequence:**

```
edit-config (org-openroadm-device/interface[name= "<XPDR<n>-NETWORK<m>-[TXRX/TX]-[λ-number]>"]/och/transmit-power) nc:operation=replace
```

set transmit power to minimum level

set transmit-power to -5 dBm

```
edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]) nc:operation=replace
```

set equipment-state of OCH interface supporting circuit-pack to not-reserved-available

```
edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]/ports[port-name= "<xponder X network port>"]) nc:operation=replace
```

set admin state of network port to outOfService

```
Get-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X network circuit pack>"]/ports[port-name= "<xponder X network port>"]/interfaces)
```

Check interfaces configured on the line port(s) (either 1 bidirectional or 2 unidirectional ports)

#### **If OCH interface is unique and bidirectional**

```
edit-config (org-openroadm-device/interface[name= "<ODU4-XPDR<n>-NETWORK<m>-TXRX>"]) nc:operation=delete
```

delete an ODU4 interface

```
edit-config (org-openroadm-device/interface[name= "<OTU4-XPDR<n>-NETWORK<m>-TXRX"]) nc:operation=delete
```

delete an OTU4 interface

```
edit-config (org-openroadm-device/interface[name= "<XPDR<n>-NETWORK<m>-[TXRX]-[λ-number] >"]) nc:operation=delete
```

delete an OCH interface XPDX-LINEN-TXRX-[λnumber]

#### **If 2 OCH unidirectional interfaces are present**

```
edit-config (org-openroadm-device/interface[name= "<ODU4-XPDR<n>-NETWORK<m>-TX>"]) nc:operation=delete
```

delete first ODU4 interface

```
edit-config (org-openroadm-device/interface[name= "<ODU4-XPDR<n>-NETWORK<m>-RX>"]) nc:operation=delete
```

```

delete second ODU4 interface
edit-config (org-openroadm-device/interface[name= "<OTU4-XPDR<n>-NETWORK<m>-TX>"])
nc:operation=delete
delete first OTU4 interface
edit-config (org-openroadm-device/interface[name= "<OTU4-XPDR<n>-NETWORK<m>-RX>"])
nc:operation=delete
delete second OTU4 interface
edit-config (org-openroadm-device/interface[name= "<XPDR<n>-NETWORK<m>-[TX]-[λ-
number] >"]) nc:operation=delete
delete first OCH interface
edit-config (org-openroadm-device/interface[name= "<XPDR<n>-NETWORK<m>-[RX]-[λ-
number] >"]) nc:operation=delete
delete second OCH interface
edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X client circuit pack>"])
nc:operation=replace
set equipment-state of 100GE interface supporting circuit-pack to not-reserved-available
Get-config (org-openroadm-device/interface[supporting-circuit-pack-name="<xponder X client circuit
pack >" and supporting-port-name="<xponder X client port >"])
Check interfaces configured on the client port(s) (either 1 bidirectional or 2 unidirectional ports)
edit-config (org-openroadm-device/circuit-packs[circuit-pack-name= "<xponder X client circuit
pack>"]/ports[port-name= "<xponder X client port>"]) nc:operation=replace
set admin state of 100GE interface supporting port to outOfService

```

***If client interface is unique and bidirectional***

```

edit-config (org-openroadm-device/interface[name= "<100GE-XPDR<n>-NETWORK<m>-
TXRX>"]) nc:operation=delete

```

***If 2 OCH unidirectional interfaces are present***

```

edit-config (org-openroadm-device/interface[name= "<100GE-XPDR<n>-NETWORK<m>-TX >"])
nc:operation=delete
edit-config (org-openroadm-device/interface[name= "<100GE-XPDR<n>-NETWORK<m>-RX >"])
nc:operation=delete

```

**End of sequence**

### 7.1.7 Add-link deletion

Configures ports, deletes connections and interfaces for an add link from a SRG-PP to a Degree TTP logical point.

***Input parameters:***

- node-id (SRGX, DEGY)
- circuit-pack (SRGX-CircuitPack, DEGY-CircuitPack)
- supporting-port (SRGX-PPN, DEGY-TTP)
- wavelength-number

- calculated-output-power
- interface (SRGX-PPN-[TXRX/RX]-[λ-number]
- DEGY-TTP-[TXRX/TX]-[λ-number])
- connection SRGX-PPN-[TXRX/RX]-[λ-number]-to-DEGY-TTP-[TXRX/TX]-[λ-number]

**Main actions sequence:**

```
edit-config (org-openroadm-device/roadm-connections/[connection-number= "<SRGX-PPN-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=replace
```

Set connection power from SRGX-PPN-[TXRX/RX] -[λ-number] to DEGY-TTP-[TXRX/TX]-[λ-number] to the minimum acceptable value:

```
set target-output-power = -60dBm
```

```
edit-config(org-openroadm-device/circuit-packs/[circuit-pack-name= "<SRGX CircuitPack >"]/ports /[/port-name= "<SRGX-PPN>"]/) nc:operation=replace
```

set admin state of OCH interface supporting PP port to outOfService

```
edit-config (org-openroadm-device/roadm-connections/[connection-number= "<SRGX-PPN-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-[λnumber]>"])nc:operation=delete
```

```
delete connection SRGX-PPN-[TXRX/RX]-[λnumber]-to- DEGY-TTP-[TXRX/TX]-[λnumber]
```

```
Get-config (org-openroadm-device/roadm-connections/)
```

check if an DEGY-TTP-[TXRX/RX]-[λ-number]-to-SRGX-PPN-[TXRX/TX]-[λ-number] connection is present

**If (SRGX-PPN & DEGY-TTP are unidirectional OR**

**NO DEGY-TTP-[TXRX/RX]-[λ-number]-to-SRGX-PPN-[TXRX/TX]-[λ-number] connection present)**

```
edit-config (org-openroadm-device/interface/[name= "<SRGX-PPN-[TXRX/RX]-[λ-number] >"]) nc:operation=delete
```

delete an OCH interface on supporting SRGX PP port

```
edit-config (org-openroadm-device/interface/[name= "<DEGY-TTP-[TXRX/TX]-[λ-number] >"]) nc:operation=delete
```

delete an OCH interface on supporting DEGY TTP port

**End of sequence**

### 7.1.8 Express link deletion

Configures the ports and deletes the interfaces and the connections for an express link between 2 Degree TTP logical points. Express link in this example is deleted in 1 step. Interfaces can be either bi (TXRX) or uni-directional (RX or TX) so that when we state [TXRX/TX] , [TXRX/RX] or [TXRX/RX/TX], only one (TXRX) or the other option (TX or RX) is valid depending on the type of interface . If the express link connections were deleted in 2 steps, then in the case of bi-directional OCH interfaces, we would delete these last-only in the second step.

**Input parameters:**

- node-id (DEGX, DEGY)
- circuit-pack (DEGX CircuitPack, DEGY CircuitPack)
- supporting-port (DEGX-TTP, DEGY-TTP)

- connection DEGX-TTP-[TXRX/ RX]-[λ-number]-to-DEGY-TTP-[TXRX/ TX]-[λ-number]

**Main actions sequence:**

```
edit-config (org-openroadm-device/roadm-connections/[connection-number= "<DEGX-TTP-[TXRX/RX]-[λnumber]-to-DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=merge
```

Set Connection DEGX-TTP-[TXRX/ RX]-[λ-number]-to-DEGY-TTP-[TXRX/ TX]-[λ-number] to low power  
set target-output-power = -60dBm

```
edit-config (org-openroadm-device/roadm-connections/[connection-number= "<DEGX-TTP-[TXRX/RX]-[λnumber]-to-DEGY-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=delete
```

Delete connection DEGX-TTP-[TXRX/RX]-[λ-number]-to-DEGY-TTP-[TXRX/TX]-[λ-number]

```
edit-config (org-openroadm-device/roadm-connections/[connection-number= "<DEGY-TTP-[TXRX/RX]-[λnumber]-to-DEGX-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=merge
```

Set Connection DEGY-TTP-[TXRX/RX]-[λ-number]-to-DEGX-TTP-[TXRX/TX]-[λ-number] to low power  
set target-output-power = -60dBm

```
edit-config (org-openroadm-device/roadm-connections/[connection-number= "<DEGY-TTP-[TXRX/RX]-[λnumber]-to-DEGX-TTP-[TXRX/TX]-[λnumber]>"]) nc:operation=delete
```

Delete connection DEGY-TTP-[TXRX/RX]-[λ-number]-to-DEGX-TTP-[TXRX/TX]-[λ-number]

```
edit-config (org-openroadm-device/interface/[name= "<DEGX-TTP-[TXRX/TX/RX]-[λ-number]>"]) nc:operation=delete
```

delete OCH interface(s) on supporting DEGX TTP port

```
edit-config (org-openroadm-device/interface/[name= "<DEGY-TTP-[TXRX/TX/RX]-[λ-number] >"]) nc:operation=delete
```

delete OCH interface(s) on supporting DEGY TTP port

**End of sequence**

### 7.1.9 Drop link deletion

Configures ports, deletes connections and interfaces for drop link between a TTP and a PP logical points.

**Input parameters:**

- node-id (DEGX, SRGY)
- circuit-pack (DEGX CircuitPack, SRGYCircuitPack)
- supporting-port (DEGX-TTP, SRGY-PPM)
- connection DEGX-TTP-[TXRX/ RX]-[λ-number]-to-SRGY-PPN-[TXRX/ TX]-[λ-number]

**Main actions sequence:**

```
edit-config(org-openroadm-device/circuit-packs/[circuit-pack-name= "<SRGY CircuitPack >"]/ports/[port-name= "<SRGY-PPM>"]) nc:operation=replace
```

set admin state of of OCH interface supporting PP port to outOfService

```
edit-config (org-openroadm-device/roadm-connections/[connection-number="<DEGX-TTP-[TXRX/RX]-[λnumber]-to- SRGY-PPM-[TXRX/TX]-[λnumber]>"]) nc:operation=delete
```

delete connection DEGX-TTP-[TXRX/RX]-[λ-number]-to-SRGY-PPN-[TXRX/TX]-[λ-number]

Get-config (org-openroadm-device/roadm-connections/)

checks if an SRGY-PPN-[TXRX/RX]-[λ-number]- to-DEGX-TTP-[TXRX/TX]-[λ-number] connection is present

**If (SRGY-PPN & DEGX-TTP are unidirectional OR NO SRGY-PPN-[TXRX/ RX]-[λ-number]- to-DEGX-TTP-[TXRX/ TX]-[λ-number] connection present)**

edit-config (org-openroadm-device/interface/[name= "<DEGX-TTP-[TXRX/RX]-[λ-number] >"])  
nc:operation=delete

delete an OCH interface on supporting DEGX TTP port

edit-config (org-openroadm-device/interface/[name= "<SRGY-PPN-[TXRX/TX]-[λ-number] >"])  
nc:operation=delete

delete an OCH interface on supporting SRGY PP port

**End of sequence**

#### 7.1.10 Note on State Models

Open ROADM Version 1.2.1 does not specify a specific state model behavior with respect to equipment and interfaces. Thus, there may be state model dependencies

- E.g., port state interaction between ports and interfaces such that a port cannot be placed in administrative down unless the supported interfaces are put in administrative down state) by a vendor's implementation.
- E.g., an entity may not be allowed to be deleted unless the administrative state is down

The equipment state model (reserved-for-facility-planned, not-reserved-planned, reserved-for-maintenance-planned, etc.) is managed by the controller and stored on the device. The actual equipment-state may or may not drive behavior on the Open ROADM device.

## 7.2 INCREMENTAL HARDWARE

### 7.2.1 Degree Growth

Degrees are to be deployed from degree number 1 to degree number N sequentially, up to the maximum number of degrees supported by the NE. This means that degree ***m*** would not be deployed before degree (***m-1***).

The growth of degrees are independent of the number of SRGs currently deployed.

In order to deploy a new degree, the following provision would take place:

- Provision new shelves as necessary
- Provision new circuit packs that will support the new degrees, and configure the ports associated with the circuit packs
  - Including setting the TTP and CTP logical connection point attributes (against the port entities)
- Provision the degree container for the new degree which include:
  - The list of circuit packs associated with the new degree.
  - The list of (external) connection ports associated with the degree

- Provision the new physical links associated with the degree. This may include:
  - Physical links internal to the degree, including the physical link for the OSC jumper that connects to the OSC pluggable circuit pack.
  - Physical links connecting the new degree to the existing degrees
  - Physical links connecting the new degree to the existing SRGs
  - Physical links to the OTDR unit

#### 7.2.1.1 Create Entities

Creates new shelf entities associated with a degree augmentation, if required.

##### **Input parameters:**

- node-id
- shelf-name
- shelf-type
- rack location
- shelf position within the rack
- administrative state
- equipment state
- due-date

Node-id, shelf-name and shelf-type are mandatory. All other parameters are optional.

- Vendors define the type of shelf based on the shelf-type.
- The rack and shelf-position identifies the location of the shelf within an office. The rack is the rack identifier, and the shelf-position identifies the location of the shelf within the rack. The naming scheme for these identifiers are up to the operator.
- Equipment-state tracks the lifecycle states of the equipment. It takes on the following values:
  - reserved-for-facility-planned: equipment is planned for use by a service
  - not-reserved-planned: equipment is planned by not reserved for any purpose
  - reserved-for-maintenance-planned: equipment is planned for use as a maintenance spare
  - reserved-for-facility-unvalidated: equipment is reserved for use by a service but not validated against planned equipment
  - not-reserved-unvalidated: equipment is not reserved for any purpose and not validated against planned equipment
  - unknown-unvalidated: unknown equipment not validated against planned equipment
  - reserved-for-maintenance-unvalidated: equipment is to be used for use as a maintenance spare but not validated against planned equipment
  - reserved-for-facility-available: reserved for use by a service and available
  - not-reserved-available: not reserved for use by a service and available



- reserved-for-maintenance-available: reserved as a maintenance spare and available
- reserved-for-reversion-inuse: equipment that is reserved as part of a home path for a service that has been temporarily re-routed
- not-reserved-inuse: equipment in use for a service
- reserved-for-maintenance-inuse: maintenance spare equipment that is in use as a maintenance spare

Read-only data:

- Physical inventory data is provided for the shelf including the vendor, model, serial-id, type (typically should match the shelf-type), product-code, manufacture-date, CLEI and hardware version.
- The slots container identifies the slots on a shelf that can host circuit-packs. The label field would be optionally present to represent the silkscreen of the slot if the slot-name does not match the silkscreen. The provisioned-circuit-pack would be present if there is a circuit pack provisioned that is plugged into the slot.

[edit-config \(/org-openroadm-device/shelves\) nc:operation=create](#)

Set the shelf-type and other shelf attributes.

Set the shelf-name

Set the shelf-type to the vendor-specific shelf type

Set the administrative-state of the shelf to inService

Set the rack, shelf-position, equipment-state and due date, if desired. These attributes are for tracking purposes only.

## End of sequence

### 7.2.1.2 Create Circuit-Pack Entities

Creates new circuit-pack entities associated with a degree augmentation.

#### **Input parameters:**

- node-id
- circuit-pack-name
- circuit-pack-type
- circuit-pack-product-code
- circuit-pack-mode
- shelf
- slot
- subSlot
- parent circuit-pack-name
- parent cp-slot-name
- administrative state
- equipment state

- due-date

Node-id, circuit-pack-name, circuit-pack-type, shelf, and slot are mandatory. All other parameters are optional.

- Vendors define the type of circuit pack based on the circuit-pack-type (mandatory) and circuit-pack-product-code (optional). The combination of these two attributes (type, product code) should uniquely identify a circuit-pack hardware.
- If a circuit pack hardware can take on different “personalities”, then this is provisioned using the circuit-pack-mode attribute (e.g., REGEN mode vs TRANSPONDER mode).
- Circuit packs refer to their location in the shelf via the shelf, slot and optional subSlot attribute. SubSlot should be set the same value as the cp-slot-name in the parent-circuit-pack container. In general, the hierarchical location of circuit packs should use the shelf and slot fields for circuit packs that plug directly into a shelf (in this case, there will not be a parent-circuit-pack container), or use the cp-slot and parent-circuit-pack fields for circuit packs that plug into other circuit packs (e.g., pluggable optics that plug into a circuit pack).
- If circuit packs are plugged into a parent circuit pack, then the parent-circuit-pack container is mandatory. In this case, the parent is specified by providing both the parent’s circuit-pack-name and cp-slot-name for the cp-slot that the circuit pack is plugging into. For example:

```
"circuit-packs": [
  {
    "circuit-pack-name": "1/1",
    "circuit-pack-type": "CPTYPE1",
    "shelf": "1",
    "slot": "1",
  },
  {
    "circuit-pack-name": "1/1/1",
    "circuit-pack-type": "CPTYPE2",
    "shelf": "1",
    "slot": "1",
    "subSlot": "1",
    "parent-circuit-pack": {
      "circuit-pack-name": "1/1"
      "cp-slot-name": "1"
    }
  },
]
```

- Equipment-state tracks the lifecycle states of the equipment. The values are the same as indicated for the shelf.

Read-only data:

- Physical inventory data is provided for the circuit-pack including the vendor, model, serial-id, type (typically should match the circuit-pack-type), product-code (typically should match the circuit-pack-product-code), manufacture-date, CLEI and hardware version.
- Circuit-pack-category provides additional classification of the circuit-pack by the vendor.
- The cp-slots container identifies the “slots” on a circuit-pack that can host other circuit-packs. The label field would be optionally present to represent the silkscreen of the cp-slot if the slot-name does not match the silkscreen. The provisioned-circuit-pack would be present if there is a child circuit pack provisioned that is plugged into the cp-slot.

[edit-config \(/org-openroadm-device/circuit-packs\) nc:operation=create](#)

Set the circuit-pack-type and other circuit-pack attributes.

Set the circuit-pack-name

Set the circuit-pack-type (mandatory) and circuit-pack-product-code (optional) to the vendor-specific values

Set the circuit-pack-mode as necessary based on vendor-specific values

Set the shelf, slot and subSlot to identify the location of the circuit-pack.

Also set the parent-circuit-pack circuit-pack-name and cp-slot-name if this circuit-pack plugs into another circuit pack.

Set the administrative-state of the shelf to inService

Set the equipment-state and due date if desired. These attributes are for tracking purposes only.

## End of sequence

### 7.2.1.2.1 Configure Circuit-Pack Port Entities

This substep configures the port entities associated with the circuit pack. Ports are auto-created upon creation of the circuit-pack.

Note: Recommendation to use the “merge” operation on editing port attributes.

#### ***Input parameters:***

- node-id
- circuit-pack-name
- port-name
- port-type
- port-qual
- circuit-id
- administrative-state
- logical-connection-point
- For OTDR ports, launch-cable-length and port-direction

Node-id, circuit-pack-name and port-name are mandatory. All other parameters are optional.

- Vendors define the port-type for the port. This value is not standardized in the Open ROADM specifications. This field is optional.
- For ROADM degree, the port-qual can take the value of roadm-external, roadm-internal, otdr or the attribute is not present. The general guideline is that the ROADM network port would be set to roadm-external and any internal ports on the data path would be set to roadm-internal. The ports on an OTDR are set to otdr. For all other ports, the port-qual is not present or optionally set to roadm-internal.
- Circuit-id is a user's defined field to identify the circuit that may be associated with this port. Its use is optional.
- The logical connection point is used to support the mapping at the controller level between the device model and network model.
  - For the ROADM degree network port, the logical connection point should be set to the format "DEG<n>-TTP-RX", "DEG<n>-TTP-TX" or "DEG<n>-TTP-TXRX" depending on if the port-direction is unidirectional receive, unidirectional transmit, or bidirectional, respectively, where <n> is the degree number. For example for degree 1 with unidirectional ports, there should be two ports – one with logical connection point = "DEG1-TTP-RX" and the other with logical connection point = "DEG1-TTP-TX"
  - For the ROADM degree internal ports that connect to another degree or SRG, the logical connection point should be set to the format "DEG<n>-CTP-RX", "DEG<n>-CTP-TX" or "DEG<n>-CTP-TXRX" depending on if the port-direction is unidirectional receive, unidirectional transmit, or bidirectional, respectively, where <n> is the degree number. For example for degree 1 with unidirectional ports that connect to degree 2, there should be two internal ports – one with logical connection point = "DEG1-CTP-RX" and the other with logical connection point = "DEG1-CTP-TX"
- For OTDR ports, the OTDR launch cable length sets the length of the launch cable and the port-direction indicates if port is associated with the degree receive or transmit port. Currently, Open ROADM runs the OTDR over the degree receive port.

#### Read-only data:

- Port-wavelength-type is set to multi-wavelength for the ROADM network degree ports. This attribute may not be present for other ports.
- Port-direction indicates if the port is modelled as a unidirectional or bidirectional port, and if unidirectional, whether the port is transmit or receive. The values for port-direction are "tx", "rx" or "bidirectional"
- Label is meant to convey the silkscreen on the device to aid in physically locating the port on the device (e.g., provided to the local technician for troubleshooting). If the port-name matches the silkscreen, then label is not necessary and may be omitted by the vendor. Otherwise, the label should be present.
- Supported-interface-capability is used to indicate which interface(s) can be built on the port. The ROADM network port list of interface capabilities include: if-OTS, if-OMS and if-OCH for ROADM degree.
- The partner-port identifies the associated port when the ports are unidirectional.

- The interface container contains a list of provisioned interfaces associated with the port.
- The roadm-port capabilities details the minimum and maximum aggregate powers supported by the port. This field applies to the ROADM network port for the degree.

**Main actions sequence:**

`edit-config (/org-openroadm-device/circuit-packs[circuit-pack-name]/ports) nc:operation=merge`

Edit the port configuration as necessary.

Set the port-name

Set the port-type to the vendor-specific port type

Set the port-qualifier and logical-connection-point as above. Note that some ports may not have these attributes set.

Set the administrative-state of the port to inService

For OTDR ports, set the launch-cable-length and port-direction.

**End of sequence**

**7.2.1.3 Create Degree Entity**

Creates new degree entity that defines the circuit-packs associated with the degree, and the connection ports for the external connections.

**Input parameters:**

- node-id
- degree-number
- list of circuit-packs associated with the degree
- list of connection-ports associated with the degree (external ports)
- For OTDR application, the OTDR's circuit-pack-name and port-name that is used to monitor this degree

Node-id, degree-number, circuit-pack-list, and connection-port-list are mandatory. All other parameters are optional.

- The circuit-packs container lists all circuit-packs associated with the degree.
  - Note that if there are shared circuit-packs that is used for more than one degree (or SRG), then that circuit-pack should also be listed under the new degree as well as the old degree (or SRG).
- The connection-ports container lists the ports that connect externally to the line transmission fibers.
  - The connection ports should refer to the physical connection to the external line fibers. This may also be the logical connections where the OTS interfaces are constructed.

- The otdr-port container provides the OTDR port that will monitor the degree. The OTDR should monitor the degree transmit port. The circuit-pack-name and port-name refer to the port on an OTDR.

Read-only data:

- The max-wavelength identifies the maximum number of wavelengths supported by the degree. This value should be set to 96.

**Main actions sequence:**

`edit-config (/org-openroadm-device/degree) nc:operation=create`

Set the degree attributes.

Set the degree-number

Set the list of circuit-packs associated to the degree

Set the list of connection ports associated to the degree

Configure the otdr-port that would monitor the degree

**End of sequence**

**7.2.1.4 Create Physical Link Entities**

Creates the physical link entities that describe the internal ROADM fibering and/or cabling within the degree, between the degree and common units, between the degree and other existing degrees, and between the degree and SRGs.

**Input parameters:**

- node-id
- physical-link-name
- source and destination port identifiers

Node-id, physical-link-name, source circuit-pack-name, source port-name, destination circuit-pack-name and destination port-name are mandatory.

- Physical links are always unidirectional
  - Source should indicate the TX port
  - Destination should indicate the RX port
- Physical links can represent a single bidirectional cable (e.g., Ethernet cable). In this case, the physical link is still modeled as two unidirectional entities.
- Physical links can also represent MPO connections. Both the physical connector and the logical connectors (child ports within the MPO) can be modeled

**Main actions sequence:**

`edit-config (/org-openroadm-device/ physical-link) nc:operation=create`

Set the physical link end points.

Set the physical-link-name

Set the physical link source circuit-pack-name and port-name (TX)  
Set the physical link destination circuit-pack-name and port-name (RX)

## End of sequence

*This completes the commissioning of a new degree.*

### 7.2.2 SRG Growth

SRGs are the add/drop units on the ROADM device. SRGs are to be deployed from SRG number 1 to SRG number N sequentially, up to the maximum number of SRGs supported by the NE. This means that SRG m would not be deployed before SRG (m-1).

The growth of SRGs are independent of the number of degrees currently deployed.

The deployment of SRGs is similar to the deployment of degrees. In order to deploy a new SRG, the following provision would take place:

- Provision new shelves as necessary
- Provision new circuit packs that will support the new SRGs, and configure the ports associated with the circuit packs
  - Including setting the TTP and CTP logical connection point attributes (against the port entities)
- Provision the shared-risk-group container for the new SRG which include:
  - The list of circuit packs associated with the new SRG.
- Provision the new physical links associated with the SRG. This may include:
  - Physical links internal to the SRG
  - Physical links connecting the new SRG to the existing degrees
  - Physical links connecting the new SRG to the existing SRGs

#### 7.2.2.1 Create Shelf Entities

Creates new shelf entities associated with a SRG augmentation, if required.

#### **Input parameters:**

- node-id
- shelf-name
- shelf-type
- rack location
- shelf position within the rack
- administrative state
- equipment state
- due-date

Node-id, shelf-name and shelf-type are mandatory. All other parameters are optional.

Refer to “Create Shelf Entities” in the degree section, Section 7.2.1.1, for a description of the provisionable and read-only attributes for shelves.

**Main actions sequence:**

`edit-config (/org-openroadm-device/shelves) nc:operation=create`

Set the shelf-type and other shelf attributes.

Set the shelf-name

Set the shelf-type to the vendor-specific shelf type

Set the administrative-state of the shelf to inService

Set the rack, shelf-position, equipment-state and due date, if desired. These attributes are for tracking purposes only.

**End of sequence**

**7.2.2.2 Create Circuit-Pack Entities**

Creates new circuit-pack entities associated with a SRG augmentation.

**Input parameters:**

- node-id
- circuit-pack-name
- circuit-pack-type
- circuit-pack-product-code
- circuit-pack-mode
- shelf
- slot
- subSlot
- parent circuit-pack-name
- parent cp-slot-name
- administrative state
- equipment state
- due-date

Node-id, circuit-pack-name, circuit-pack-type, shelf, slot are mandatory. All other parameters are optional.

Refer to “Create Circuit-Pack Entities” in the degree section, Section 7.2.1.2, for a description of the provisionable and read-only attributes for shelves.

**Main actions sequence:**

`edit-config (/org-openroadm-device/circuit-packs) nc:operation=create`

Set the circuit-pack-type and other circuit-pack attributes.

Set the circuit-pack-name



Set the circuit-pack-type (mandatory) and circuit-pack-product-code (optional) to the vendor-specific values

Set the circuit-pack-mode as necessary based on vendor-specific values

Set the shelf, slot and subSlot to identify the location of the circuit-pack.

Also set the parent-circuit-pack circuit-pack-name and cp-slot-name if this circuit-pack plugs into another circuit pack.

Set the administrative-state of the shelf to inService

Set the equipment-state and due date if desired. These attributes are for tracking purposes only.

## End of sequence

### 7.2.2.2.1 Configure Circuit-Pack Port Entities

This substep configures the port entities associated with the circuit pack. Ports are auto-created upon creation of the circuit-pack.

Note: Recommendation to use the “merge” operation on editing port attributes.

#### ***Input parameters:***

- node-id
- circuit-pack-name
- port-name
- port-type
- port-qual
- circuit-id
- administrative-state
- logical-connection-point

Node-id, circuit-pack-name and port-name are mandatory. All other parameters are optional.

Refer to “Create Shelf Entities” in the degree section, Section 7.2.1.2.1, for a description of the provisionable and read-only attributes for shelves. Differences specific to SRGs are described below.

- For ROADM SRGs, the port-qual can take the value of roadm-external, roadm-internal, or the attribute is not present. The general guideline is that the ROADM SRG add/drop port would be set to roadm-external and any internal SRG ports on the data path would be set to roadm-internal. For all other ports, the port-qual is not present or optionally set to roadm-internal.
- The logical connection point is used to support the mapping at the controller level between the device model and network model.
  - For the ROADM SRG add/drop port, the logical connection point should be set to the format “SRG<n>-PP<m>”, where <n> is the SRG number and <m> is the add/drop port pair identifier. For example for SRG 1 add/drop port #7 would have the logical connection point set to SRG1-PP7.

- For the ROADM SRG internal ports that connect to another degree or SRG, the logical connection point should be set to the format “SRG<n>-CP-RX”, “SRG<n>-CP-TX” or “SRG<n>-CP-TXRX” depending on if the port-direction is unidirectional receive, unidirectional transmit, or bidirectional, respectively, where <n> is the SRG number. For example for SRG 1 with unidirectional ports that connect to degree 1, there should be two internal ports – one with logical connection point = “SRG1-CP-RX” and the other with logical connection point = “SRG1-CP-TX”

Read-only data:

- Port-wavelength-type is set to wavelength for the ROADM SRG add/drop port. This attribute may not be present for other ports.
- Supported-interface-capability is used to indicate which interface(s) can be built on the port. The ROADM SRG add/drop port, the list of interface capabilities include: if-OCH.
- The roadm-port capabilities details the minimum and maximum aggregate powers supported by the port. This field applies to the ROADM SRG add/drop ports.

#### **Main actions sequence:**

[edit-config \(/org-openroadm-device/circuit-packs\[circuit-pack-name\]/ports\) nc:operation=merge](#)

Edit the port configuration as necessary.

Set the port-name

Set the port-type to the vendor-specific port type

Set the port-qualifier and logical-connection-point as above. Note that some ports may not have these attributes set.

Set the administrative-state of the port to inService

#### **End of sequence**

##### **7.2.2.3 Create Shared Risk Group Entity**

Creates new shared-risk-group entity that defines the circuit-packs associated with the SRG.

#### **Input parameters:**

- node-id
- srg-number
- list of circuit-packs associated with the SRG

Node-id, srg-number, and circuit-pack-list are mandatory.

- The circuit-packs container lists all circuit-packs associated with the SRG.
  - Note that if there are shared circuit-packs that are used for more than one SRG (or degree), then that circuit-pack should also be listed under the new SRG as well as the old SRG (or degree).

Read-only data:

- The max-add-drop-ports identifies the maximum number of SRG add/drop ports supported by the SRG.

**Main actions sequence:**

`edit-config (/org-openroadm-device/shared-risk-group) nc:operation=create`

Set the SRG attributes.

Set the srg-number

Set the list of circuit-packs associated to the SRG

**End of sequence**

**7.2.2.4 Create Physical Link Entities**

Creates the physical link entities that describe the internal ROADM fiber and/or cabling within the SRG, between the SRG and common units, between the SRG and other degrees, and between the SRG and other existing SRGs.

**Input parameters:**

- node-id
- physical-link-name
- source and destination port identifiers

Node-id, physical-link-name, source circuit-pack-name, source port-name, destination circuit-pack-name and destination port-name are mandatory.

- Physical links are always unidirectional
  - Source should indicate the TX port
  - Destination should indicate the RX port
- Physical links can represent a single bidirectional cable (e.g., Ethernet cable). In this case, the physical link is still modeled as two unidirectional entities.
- Physical links can also represent MPO connections. Both the physical connector and the logical connectors (child ports within the MPO) can be modeled

**Main actions sequence:**

`edit-config (/org-openroadm-device/ physical-link) nc:operation=create`

Set the physical link end points.

Set the physical-link-name

Set the physical link source circuit-pack-name and port-name (TX)

Set the physical link destination circuit-pack-name and port-name (RX)

**End of sequence**

*This completes the commissioning of a new SRG.*

### 7.2.3 Xponder Growth

Xponders represent the transponder, muxponders and other xponder types supported by the network element.

Individual xponders instances are not identified in the model explicitly like degrees or SRGs.

Xponders have network port(s) that connects to the ROADMs' SRG add/drop ports.

Xponders have client port(s) that connects to Routers or other client equipment.

The deployment of xponders is similar to the deployment of degrees and SRGs with the exception there is not an explicit "xponder" container. In order to deploy new xponder equipment, the following provision would take place:

- Provision new shelves as necessary
- Provision new circuit packs that will support the new xponder, and configure the ports associated with the circuit packs
  - Including setting the logical connection point attributes (against the port entities)
- Provision the new physical links associated with the xponder as applicable. The physical links would be needed if the xponder had multiple circuit packs with physical cabling between the circuit-packs.

#### 7.2.3.1 Create Shelf Entities

Creates new shelf entities associated with a xponder augmentation, if required.

##### **Input parameters:**

- node-id
- shelf-name
- shelf-type
- rack location
- shelf position within the rack
- administrative state
- equipment state
- due-date

Node-id, shelf-name and shelf-type are mandatory. All other parameters are optional.

Refer to "Create Shelf Entities" in the degree section, Section 7.2.1.1, for a description of the provisionable and read-only attributes for shelves.

[edit-config \(/org-openroadm-device/shelves\) nc:operation=create](#)

Set the shelf-type and other shelf attributes.

Set the shelf-name

Set the shelf-type to the vendor-specific shelf type

Set the administrative-state of the shelf to inService

Set the rack, shelf-position, equipment-state and due date, if desired. These attributes are for tracking purposes only.

## End of sequence

### 7.2.3.2 Create Circuit-Pack Entities

Creates new circuit-pack entities associated with a xponder augmentation.

#### **Input parameters:**

- node-id
- circuit-pack-name
- circuit-pack-type
- circuit-pack-product-code
- circuit-pack-mode
- shelf
- slot
- subSlot
- parent circuit-pack-name
- parent cp-slot-name
- administrative state
- equipment state
- due-date

Node-id, circuit-pack-name, circuit-pack-type, shelf, slot are mandatory. All other parameters are optional.

Refer to “Create Circuit-Pack Entities” in the degree section, Section 7.2.1.2, for a description of the provisionable and read-only attributes for shelves.

#### **Main actions sequence:**

`edit-config (/org-openroadm-device/circuit-packs) nc:operation=create`

Set the circuit-pack-type and other circuit-pack attributes.

Set the circuit-pack-name

Set the circuit-pack-type (mandatory) and circuit-pack-product-code (optional) to the vendor-specific values

Set the circuit-pack-mode as necessary based on vendor-specific values

Set the shelf, slot and subSlot to identify the location of the circuit-pack.

Also set the parent-circuit-pack circuit-pack-name and cp-slot-name if this circuit-pack plugs into another circuit pack.

Set the administrative-state of the shelf to inService

Set the equipment-state and due date if desired. These attributes are for tracking purposes only.

## End of sequence

### 7.2.3.2.1 Configure Circuit-Pack Port Entities

This substep configures the port entities associated with the circuit pack. Ports are auto-created upon creation of the circuit-pack.

Note: Recommendation to use the “merge” operation on editing port attributes.

#### ***Input parameters:***

- node-id
- circuit-pack-name
- port-name
- port-type
- port-qual
- circuit-id
- administrative-state
- logical-connection-point

Node-id, circuit-pack-name and port-name are mandatory. All other parameters are optional.

Refer to “Create Shelf Entities” in the degree section, Section 7.2.1.2.1, for a description of the provisionable and read-only attributes for shelves. Differences specific to SRGs are described below.

- For xponders, the port-qual can take the value of xpdr-network, xpdr-client, or the attribute is not present. The general guideline is that the xponder network ports would be set to xpdr-network and xponder client ports would be set to xpdr-client. For all other ports, the port-qual is not present.
- The logical connection point is used to support the mapping at the controller level between the device model and network model.
  - For the xponder network port, the logical connection point should be set to the format XPDR<n>-NETWORK<m>.
  - For the xponder client ports, the logical connection point should be set to the format XPDR<n>-CLIENT<m>.
  - Where <n> is the logical transponder/muxponder/xponder number, and <m> are the network and client ports that make up the transponder/muxponder/xponder. Typically, the network <m> will be 1 for transponders and muxponers. For transponders, the client <m> will be 1 and for muxponders the client <m> will be 1 .. m where m = max number of client muxponder ports.

#### Read-only data:

- Port-wavelength-type is set to wavelength for the xponder network and client ports. This attribute may not be present for other ports.
- Supported-interface-capability is used to indicate which interface(s) can be built on the port.

- The xponder network port, the list of interface capabilities include: if-OCH. (Note: MSA v1.2.1 does not support an explicit identity for if-OTU4 nor if-ODU4)
  - The xponder client port, the list of interface capabilities may include: if-100GE and if-OCH. The client supports both 100GE and OTU4. (Note: MSA v1.2.1 does not support an explicit identity for if-OTU4 nor if-ODU4)
- The transponder-port capabilities details the minimum and maximum aggregate powers supported by the port. This field applies to the xponder network and client ports.

**Main actions sequence:**

`edit-config (/org-openroadm-device/circuit-packs[circuit-pack-name]/ports) nc:operation=merge`

Edit the port configuration as necessary.

Set the port-name

Set the port-type to the vendor-specific port type

Set the port-qualifier and logical-connection-point as above. Note that some ports may not have these attributes set.

Set the administrative-state of the port to inService

**End of sequence**

**7.2.3.3 Create Physical Link Entities**

Creates the physical link entities that describe the internal xponder fiber and/or cabling, if applicable.

**Input parameters:**

- node-id
- physical-link-name
- source and destination port identifiers

Node-id, physical-link-name, source circuit-pack-name, source port-name, destination circuit-pack-name and destination port-name are mandatory.

- Physical links are always unidirectional
  - Source should indicate the TX port
  - Destination should indicate the RX port
- Physical links can represent a single bidirectional cable (e.g., Ethernet cable). In this case, the physical link is still modeled as two unidirectional entities.
- Physical links can also represent MPO connections. Both the physical connector and the logical connectors (child ports within the MPO) can be modeled

**Main actions sequence:**

`edit-config (/org-openroadm-device/ physical-link) nc:operation=create`

Set the physical link end points.

Set the physical-link-name

Set the physical link source circuit-pack-name and port-name (TX)  
Set the physical link destination circuit-pack-name and port-name (RX)

### **End of sequence**

*This completes the commissioning of a new xponder unit(s).*

## **8 MAINTENANCE ACTION USE CASES**

---

### **8.1 RESET/RESTART AT SYSTEM OR CIRCUIT PACK LEVEL**

More detail on resets and restarts will be added in the next version of this Whitepaper, including sections on system-level non-service affecting reset, circuit pack level non-service affecting reset (data plane intact) and circuit pack level power cycle reset (data plane down)

### **8.2 SET OTU/ODU TTI FIELDS (SAPI/DAPI MESSAGE) FOR CONNECTIVITY MANAGEMENT**

More detail on TTI will be added in the next version of this Whitepaper, including describing the mechanism and semantics to set OTU/ODU interface TTI fields needed for connectivity management.

### **8.3 OPERATE/RELEASE LOOPBACK ON AN INTERFACE**

More detail on loopback operations will be added in the next version of this Whitepaper, including describing the mechanism and semantics to operate/release loop-backs:

Operator/Release a loop-back toward the external (fiber) connections on a (100G) client interface

Operator/Release a loop-back toward the internal connections on a (100G) client interface

Operator/Release a loop-back toward the external (fiber) connections on an OTU4 network interface

Operator/Release a loop-back toward the internal connections on an OTU4 network interface

### **8.4 GENERATE/COLLECT DEBUG DUMP FILE(S) FOR IN-DEPTH FAILURE ANALYSIS**

To be addressed in Open ROADM V2

### **8.5 RESTORE REMOTE COMMUNICATIONS TO THE DEVICE FROM CORRUPTED OR INCONSISTENT CONFIGURATION**

More detail on restoring communications to a device will be added in the next version of this Whitepaper, including describing the mechanism to restore communications to the device from corrupted or inconsistent configuration, assuming power cycle of the processor or management blade is ineffective in this case.



## 8.6 RE-SYNC WITH NORTHBOUND CONTROLLER/APPLICATIONS AFTER EXTENDED PERIOD OF COMMUNICATION LOSS

More detail on controller resync will be added in the next version of this Whitepaper, including describing the mechanism and semantics for the controller to re-sync up with the device after extended period of communication loss, to the extent that network buffer is overflow and controller has to recollect the events from device log file(s).

### 8.1 MISSING TOPICS (TO BE ADDRESSED IN FUTURE VERSION OF WHITEPAPER)

- What does “admin-state” actually means
  - Suppression of alarms only, no impact to service state
    - "inService" - Services requested to be up, alarm reporting is active, no maintenance operation allowed.
    - "outOfService" - Services requested to be up, alarms suppressed, no maintenance operation allowed.
    - "maintenance" - Services requested to be up, alarms suppressed, maintenance operation allowed.
- What failures should drive operational states
- Alarm suppression rules and guidelines
- RPCs
  - Disable automatic shutoff
  - Connection port trail
- Wavelength map descriptions
  - 0 is reserved and means “laser off”
- Min-power / max-power interpretations and usage
- Internal link usage and examples’
- Span loss changes on existing system

## 9 APPENDIX A: MANIFEST FILE FOR SOFTWARE DOWNLOAD AND DATABASE OPERATIONS

---

### 9.1 INTRODUCTION

This section describes the manifest file for software download, database backup and database restore operations. It includes details of how an Open ROADM implementation would specify the file transfer, staging, backup, restore and activation of software loads and database archives.

The manifest file allows vendor to specify vendor-specific behavior in a common format for software download and database operations. This allows the controller to automatically adjust procedures to the vendor specific requirements.

The manifest file is expected to be provided out-of-band (i.e., the file is not provided directly with the network element but provided offline by the vendor to the controller).

There may be multiple versions of the manifest file distinguished by the vendor, model and sw-version. It is expected that the database operations would apply to the current software version running on the Open ROADM device. For software download operations, the sw-version would be the software load that the operation would upgrade to (e.g., sw-version = “2.0” when the current software version is “1.2.1”). Note that the sw-version is the vendor software version and not necessarily the Open ROADM version.

#### 9.1.1 Software Download

Software download is expected to take place as a series of operations as specified in the manifest file. There should be a manifest file created for the new software load. The manifest file specifies one or more instruction-sets. Each instruction-set would apply to one or more from-sw-versions that would represent the current sw-version running on the device. This allows different instructions based on the upgrade from different current software versions.

The typical order of operations for software download is as follows:

- 1) Transfer one or more software files from an SFTP server to the device.
- 2) Stage the software file
  - a. As part of staging, additional software files may be transferred from the SFTP server to the device. The details of what software files to transfer would be vendor-dependent and the file information should be in the software file that was staged. These software files can be in a flat or hierarchical sub-directory structure on the SFTP server.
- 3) If needed, delete the software file.
- 4) Steps 1-3 may repeat if the device storage does not support transferring all software files at once.
- 5) Activate the software with an optional validation timer
- 6) Typically it is expected that the device would reboot at this time (reboot should be automatic).
- 7) When the device comes up, the operator would evaluate the software load in the validation time (if supported and specified in the manifest file).

- a. During this time, access to the database may be restricted to allow for rollback. Thus, some provisioning activities (e.g., new service activation) may be restricted.
- 8) If the software load is acceptable, the operator would cancel the validation time and accept the software load.
- 9) The device is now fully functional at the new software release

The operator may also cancel the validation and not accept the new software load. Under this case, the device will reboot and revert back to the original software load.

Note that if the device did not support a validation period, then reversion to the original software load may not be supported.

### 9.1.2 Database Backup

The database backup operation will backup the current database from the device and transfer it to an SFTP server for offline storage.

The typical order of operations for database backup is as follows:

- 1) Initiate the database backup operation and specify the filename. The device will create the database file on the device for file transfer.
- 2) When the database file is completed (either as a synchronous command or a an asynchronous command with an event notification), the controller would transfer the file from the device to the SFTP server.
- 3) The controller would then delete the database file from the device.

The database backup manifest file defines two variables: `__LOCAL-FILE-PATH` and `__REMOTE-FILENAME`

- `__LOCAL-FILE-PATH` is the local filename (and path) for the creation of the database backup file
- `__REMOTE-FILENAME` is the remote filename for storage on the SFTP server to be used in the transfer-file RPC

It is expected that the user or controller will provide the value of these attributes.

### 9.1.3 Database Restore

The database restore operation will restore a previously saved (backup) database file from an SFTP server.

The typical order of operations for database restore is as follows:

- 1) Transfer the previously backed up database file from the SFTP server to the device
- 2) Initiate a database restore operation on the device.
- 3) Once the database restore operation completes, the database file can be deleted from the device.
- 4) Activate the database restored file with an optional rollback timer (similar to the software download validation timer)
- 5) Typically it is expected that the device would reboot at this time (reboot should be automatic).

- 6) When the device comes up, the operator would evaluate the device with the restored database in the rollback time (if supported and specified in the manifest file).
  - a. During this time, access to the database may be restricted to allow for rollback. Thus, some provisioning activities (e.g., new service activation) may be restricted.
- 7) If the database restore is acceptable, the operator would cancel the rollback timer and accept the database.
- 8) The device is now fully functional at the point of the restored database file

The operator may also cancel the rollback timer and not accept the database restore. Under this case, the device will reboot and revert back to the database in the state before the database restore.

Note that if the device did not support a rollback period, then reversion to the previous database may not be supported.

The database restore manifest file defines three variables: `__LOCAL-FILE-PATH`, `__REMOTE-FILENAME`, and `__NODE-ID-CHECK`

- `__LOCAL-FILE-PATH` is the local filename (and path) where the database file will be stored on the device for restoration
- `__REMOTE-FILENAME` is the remote database filename on the SFTP server to be used in the transfer-file RPC to the device
- `__NODE-ID-CHECK` is to have the device validate that the current node-id and the node-id in the database file match to ensure the correct database file is being restored

It is expected that the user or controller will provide the value of these attributes.

## 9.2 MANIFEST FILE YANG MODULE

The format for the manifest file is provided below.

### 9.2.1 Tree View

```
module: org-openroadm-manifest-file
  +--rw sw-manifest!
  |   +--rw vendor                string
  |   +--rw model                 string
  |   +--rw sw-version            string
  |   +--rw global-async-timeout? uint16
  |   +--rw global-sync-timeout?  uint16
  |   +--rw instruction-set* [index]
  |       +--rw index                uint8
  |       +--rw from-sw-version*     string
  |       +--rw is-commit-sw-activate-async? boolean
  |       +--rw cancel-validation-timer-async-timeout? uint16
  |       +--rw cancel-validation-timer-sync-timeout?  uint16
  |       +--rw sw-manifest-commands
  |           +--rw sw-manifest-command* [command-order]
  |               +--rw command-order    uint8
  |               +--rw command          identityref
  |               +--rw download-file
  |                   | +--rw remote-filename    string
  |                   | +--rw local-file-path    string
  |                   | +--rw timeout?          uint16
  |                   | +--rw is-async?         boolean
```

```

|         +--rw delete-file
|         | +--rw filename      string
|         | +--rw timeout?     uint16
|         | +--rw is-async?    boolean
|         +--rw sw-stage
|         | +--rw filename?    string
|         | +--rw timeout?     uint16
|         | +--rw is-async?    boolean
|         +--rw sw-activate
|         | +--rw version      string
|         | +--rw validation-timer? string
|         | +--rw timeout?     uint16
|         | +--rw auto-reboot  uint16
|         | +--rw is-async?    boolean
+--rw db-backup-manifest!
| +--rw vendor                string
| +--rw model                  string
| +--rw sw-version?           string
| +--rw global-async-timeout? uint16
| +--rw global-sync-timeout?  uint16
| +--rw db-backup-manifest-commands
|   +--rw db-backup-manifest-command* [command-order]
|     +--rw command-order      uint8
|     +--rw command            identityref
|     +--rw upload-file
|     | +--rw remote-filename  string
|     | +--rw local-file-path  string
|     | +--rw timeout?         uint16
|     | +--rw is-async?        boolean
|     +--rw delete-file
|     | +--rw filename         string
|     | +--rw timeout?         uint16
|     | +--rw is-async?        boolean
|     +--rw db-backup
|     | +--rw filename?        string
|     | +--rw timeout?         uint16
|     | +--rw is-async?        boolean
+--rw db-restore-manifest!
| +--rw vendor                string
| +--rw model                  string
| +--rw sw-version?           string
| +--rw global-async-timeout? uint16
| +--rw global-sync-timeout?  uint16
| +--rw is-commit-db-activate-async? boolean
| +--rw cancel-rollback-timer-async-timeout? uint16
| +--rw cancel-rollback-timer-sync-timeout?  uint16
| +--rw database-init-sync-timeout?          uint16
| +--rw db-restore-manifest-commands
|   +--rw db-restore-manifest-command* [command-order]
|     +--rw command-order      uint8
|     +--rw command            identityref
|     +--rw download-file
|     | +--rw remote-filename  string
|     | +--rw local-file-path  string
|     | +--rw timeout?         uint16
|     | +--rw is-async?        boolean
|     +--rw delete-file
|     | +--rw filename         string
|     | +--rw timeout?         uint16
|     | +--rw is-async?        boolean
|     +--rw db-restore
|     | +--rw filename?        string
|     | +--rw node-id-check?   string
|     | +--rw timeout?         uint16
|     | +--rw is-async?        boolean
|     +--rw db-activate

```

```

+--rw rollback-timer?   string
+--rw timeout?          uint16
+--rw auto-reboot       uint16
+--rw is-async?         boolean

```

## 9.2.2 YANG module (this module will be finalized and provided as part of Open ROADM Version 2)

```

module org-openroadm-manifest-file {
  namespace "http://org/openroadm/manifest-file";
  prefix org-openroadm-manifest-file;

  organization "Open ROADM MSA";
  contact
    "OpenROADM.org";
  description
    "YANG definitions of sw-manifest-file

    Copyright of the Members of the Open ROADM MSA Agreement dated (c) 2017,
    AT&T Intellectual Property. All other rights reserved.

    Redistribution and use in source and binary forms, with or without modification,
    are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice, this
      list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright notice,
      this list of conditions and the following disclaimer in the documentation and/or
      other materials provided with the distribution.
    * Neither the Members of the Open ROADM MSA Agreement nor the names of its
      contributors may be used to endorse or promote products derived from this software
      without specific prior written permission.

    THIS SOFTWARE IS PROVIDED BY THE MEMBERS OF THE OPEN ROADM MSA AGREEMENT 'AS IS'
    AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
    WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
    IN NO EVENT THE MEMBERS OF THE OPEN ROADM MSA AGREEMENT BE LIABLE FOR ANY DIRECT,
    INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
    NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
    OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
    WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
    ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
    POSSIBILITY OF SUCH DAMAGE.

    Also contains code components extracted from IETF netconf. These code components
    are copyrighted and licensed as follows:

    Copyright (c) 2016 IETF Trust and the persons identified as the document authors.
    All rights reserved.

    This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating
    to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of
    publication of this document. Please review these documents carefully, as they
    describe your rights and restrictions with respect to this document. Code Components
    extracted from this document must include Simplified BSD License text as described in
    Section 4.e of the Trust Legal Provisions and are provided without warranty as
    described in the Simplified BSD License."

revision 2017-06-20 {
  description
    "Version 2.0 - preliminary version";
  reference "This module serves as the manifest file reference.";
}

identity manifest-commands {
  description "base identity for defining manifest-commands.";
}

identity download-file {
  base manifest-commands;
}

```

```

    description "download-file (transfer from OWB-C to Device)";
}

identity upload-file {
    base manifest-commands;
    description "upload-file (transfer from Device to OWB-C)";
}

identity delete-file {
    base manifest-commands;
    description "delete-file from device";
}

identity sw-manifest-commands {
    base manifest-commands;
    description "base identity for defining manifest-commands specific to sw-manifest.";
}

identity sw-stage {
    base sw-manifest-commands;
    description "sw-stage sw-manifest-command";
}

identity sw-activate {
    base sw-manifest-commands;
    description "sw-activate sw-manifest-command";
}

identity db-backup-manifest-commands {
    base manifest-commands;
    description "base identity for defining manifest-commands specific to db-backup-manifest.";
}

identity db-backup {
    base db-backup-manifest-commands;
    description "db-backup db-backup-manifest-command";
}

identity db-restore-manifest-commands {
    base manifest-commands;
    description "base identity for defining manifest-commands specific to db-restore-manifest.";
}

identity db-restore {
    base db-restore-manifest-commands;
    description "db-restore db-restore-manifest-command";
}

identity db-activate {
    base db-restore-manifest-commands;
    description "db-activate db-restore-manifest-command";
}

grouping base-manifest {
    description "base set of variables in all manifest files";
    leaf vendor {
        type string;
        mandatory true;
        description "This field should match the /org-openroadm-device/info/vendor.
            It is assumed that the vendor value does not change during the
            processing of the manifest file.

            The controller agent would use the vendor and model to find the
            manifest for an Open ROADM NE. The controller agent would also
            use the vendor and model to validate that this is a valid manifest
            for the Open ROADM NE.
            ";
    }
    leaf model {
        type string;
        mandatory true;
    }
}

```

```

        description "This field should match the /org-openroadm-device/info/model.
            It is assumed that the model value does not change during the
            processing of the manifest file.

            The controller agent would use the vendor and model to find the
            manifest for an Open ROADM NE. The controller agent would also
            use the vendor and model to validate that this is a valid manifest
            for the Open ROADM NE.
        ";
    }
    leaf sw-version {
        type string;
        description "This field should match the
            /org-openroadm-device/info/softwareVersion.
            This is the value in the info tree AFTER an upgrade.
        ";
    }
    leaf global-async-timeout {
        type uint16;
        default 900; // 15 minutes
        description "global-async-timeout - time in seconds to wait for command processing to
            complete before polling the device to determine the result.

            This global-async-timeout applies to any asynchronous command.
        ";
    }
    leaf global-sync-timeout {
        type uint16;
        description "global-sync-timeout - time in seconds to wait for the rpc response for
            synchronous commands.

            This global-sync-timeout applies to any synchronous command.

            No default is modeled; if not provided, defaults to the global
            timeout supported by the controller for rpc responses.
        ";
    }
}

container sw-manifest {
    presence "The sw-manifest instructions for swdl operations have been defined.";
    description "The manifest file provides instructions to a software download
        application to download and install a new software load into a vendor's
        equipment.

        Software download files
        All vendor files for a software release should be stored in a
        separate directory. A unique directory would be used for each vendor,
        model and software release combination. This directory and all files in
        that directory will be accessible by the SFTP server.
        The software directory can be flat or hierarchical with
        subdirectories. The manifest file should be in the root directory of the
        software directory.
        A software directory must contain files for one and only one
        software release.

        Manifest file name
        Each software release directory shall contain a manifest file for
        that release. The filename for the manifest file shall be sw-manifest.json.
    ";
    uses base-manifest {
        refine sw-version {
            mandatory true;
        }
    }
    list instruction-set {
        key "index";
        leaf index {
            type uint8;
            description "The index for this instruction set.";
        }
    }
}

```



```

description "The instruction set for a list of sw-versions that can be upgraded to
the sw-version specified at the top of the manifest file.";
leaf-list from-sw-version {
  type string;
  description "The optional list of sw-versions that can be upgraded to the
sw-version specified at the top of the sw-manifest file.

If not specified, this instruction set is used to upgrade from
any sw-version to the sw-version specified at the top of the
sw-manifest file.

If multiple instruction sets are provided, from-sw-version
should always be defined.";
}
leaf is-commit-sw-activate-async {
  type boolean;
  default "true";
  description "Is cancel-validation-timer (accept = true) supported as an
async or sync command on the device? If supported as sync, the rpc response
is used to determine success/failure instead of waiting for transient notifications
of the result.
NOTE: cancel-validation-timer (accept = false) requires a reboot so is
always considered async";
}
leaf cancel-validation-timer-async-timeout {
  type uint16;
  description "timeout value to use for cancel-validation-timer when supported as
an async command. If not specified, the global-async-timeout is used.";
}
leaf cancel-validation-timer-sync-timeout {
  type uint16;
  description "timeout value to use for cancel-validation-timer (accept = true) when
supported as a sync command. If not specified, the global-sync-timeout
is used.";
}
container sw-manifest-commands {
  description "The ordered list of commands to be processed. Since some yang
implementations do not support ordered-by user, the list is also
indexed by command-order. The commands should be processed
in the order of command-order.

Processing moves to the next command when:
1. command is synchronous and rpc returns a successful result.
2. command is asynchronous, the rpc returns a successful result,
and
2.1 expected successful notification(s) have been received; or
2.2 timeout occurs and polling determines a successful result.

Processing of the manifest file is aborted when:
1. command is synchronous and rpc returns a failed result.
2. command is asynchronous, and:
2.1 the rpc returns a failed result; or
2.2 a failed notification is received; or
2.3 timeout occurs and polling determines a failed result.

NOTE: behavior for timeouts (synchronous or asynchronous) may depend upon
controller implementation per command. It may be considered either:
- as a successful result
- as a failed result
- as a success or failure based on polling the device
";
  list sw-manifest-command {
    key "command-order";
    ordered-by user;
    description "The list of commands to be processed.";
    leaf command-order {
      type uint8;
      description "The order in which commands should be processed.";
    }
    leaf command {
      type identityref {

```

```

        base sw-manifest-commands;
    }
    mandatory true;
    description "The command to be processed.";
}
uses download-file-command;
uses delete-file-command;
uses sw-stage-command;
uses sw-activate-command;
}
}
}
}

container db-backup-manifest {
    presence "The db-backup-manifest template for db-backup operations has been defined.";
    description "The manifest file provides instructions to a database operations
        application to backup the database on a device.

        Since the files used for these operations are likely user selected,
        these manifest files are more likely used by the controller as a
        template to control the overall flow of a backup operation and provide
        a means of providing customized timeout values.

        The following strings will be recognized as parameters to be replaced
        by the user selected values: __LOCAL-FILE-PATH, __REMOTE-FILENAME.

        Manifest file name
        Each vendor/model combination can have a separate manifest file
        defined for backup. These shall be named db-backup-manifest.json.
    ";
    uses base-manifest;
    container db-backup-manifest-commands {
        description "The ordered list of commands to be processed. Since some yang
            implementations do not support ordered-by user, the list is also
            indexed by command-order. The commands should be processed
            in the order of command-order.

            Processing moves to the next command when:
            1. command is synchronous and rpc returns a successful result.
            2. command is asynchronous, the rpc returns a successful result,
               and
               2.1 expected successful notification(s) have been received; or
               2.2 timeout occurs and polling determines a succesful result.

            Processing of the manifest file is aborted when:
            1. command is synchronous and rpc returns a failed result.
            2. command is asynchronous, and:
               2.1 the rpc returns a failed result; or
               2.2 a failed notification is received; or
               2.3 timeout occurs and polling determines a failed result.

            NOTE: behavior for timeouts (synchronous or asynchronous) may depend upon
            controller implementation per command. It may be considered either:
            - as a successful result
            - as a failed result
            - as a success or failure based on polling the device
        ";
        list db-backup-manifest-command {
            key "command-order";
            ordered-by user;
            description "The list of commands to be processed.";
            leaf command-order {
                type uint8;
                description "The order in which commands should be processed.";
            }
            leaf command {
                type identityref {
                    base db-backup-manifest-commands;
                }
            }
            mandatory true;

```

```

        description "The command to be processed.";
    }
    uses upload-file-command;
    uses delete-file-command;
    uses db-backup-command;
}
}
}

container db-restore-manifest {
    presence "The db-restore-manifest template for db-restore operations has been defined.";
    description "The manifest file provides instructions to a database operations
        application to restore the database on a device.

        Since the files used for these operations are likely user selected,
        these manifest files are more likely used by the controller as a
        template to control the overall flow of a restore operation and provide
        a means of providing customized timeout and auto-reboot values.

        The following strings will be recognized as parameters to be replaced
        by the user selected values: __LOCAL-FILE-PATH, __REMOTE-FILENAME,
        __NODE-ID-CHECK.

        Manifest file name
            Each vendor/model combination can have a separate manifest file
            defined for restore. These shall be named db-restore-manifest.json.
        ";
    uses base-manifest;
    leaf is-commit-db-activate-async {
        type boolean;
        default "true";
        description "Is cancel-rollback-timer (accept = true) supported as an
            async or sync command on the device? If supported as sync, the rpc response
            is used to determine success/failure instead of waiting for transient notifications
            of the result.
            NOTE: cancel-rollback-timer (accept = false) requires a reboot so is
            always considered async";
    }
    leaf cancel-rollback-timer-async-timeout {
        type uint16;
        description "timeout value to use for cancel-rollback-timer when supported as
            an async command. If not specified, the global-async-timeout is used.";
    }
    leaf cancel-rollback-timer-sync-timeout {
        type uint16;
        description "timeout value to use for cancel-rollback-timer (accept = true) when
            supported as a sync command. If not specified, the global-sync-timeout
            is used.";
    }
    leaf database-init-sync-timeout {
        type uint16;
        description "timeout value to use for database-init command. If not specified,
            the global-sync-timeout is used.";
    }
}

container db-restore-manifest-commands {
    description "The ordered list of commands to be processed. Since some yang
        implementations do not support ordered-by user, the list is also
        indexed by command-order. The commands should be processed
        in the order of command-order.

        Processing moves to the next command when:
        1. command is synchronous and rpc returns a successful result.
        2. command is asynchronous, the rpc returns a successful result,
            and
            2.1 expected successful notification(s) have been received; or
            2.2 timeout occurs and polling determines a succesful result.

        Processing of the manifest file is aborted when:
        1. command is synchronous and rpc returns a failed result.
        2. command is asynchronous, and:
            2.1 the rpc returns a failed result; or

```

- 2.2 a failed notification is received; or
- 2.3 timeout occurs and polling determines a failed result.

NOTE: behavior for timeouts (synchronous or asynchronous) may depend upon controller implementation per command. It may be considered either:

- as a successful result
- as a failed result
- as a success or failure based on polling the device

```

";
list db-restore-manifest-command {
  key "command-order";
  ordered-by user;
  description "The list of commands to be processed.";
  leaf command-order {
    type uint8;
    description "The order in which commands should be processed.";
  }
  leaf command {
    type identityref {
      base db-restore-manifest-commands;
    }
    mandatory true;
    description "The command to be processed.";
  }
  uses download-file-command;
  uses delete-file-command;
  uses db-restore-command;
  uses db-activate-command;
}
}

grouping timeout-command {
  description "timeout-command is to be used by any manifest command supporting a timeout";
  leaf timeout {
    type uint16;
    description "See command for additional details.
      if command is async,
        - overrides the global-async-timeout;
        - defaults to the global-async-timeout if not provided.
      if command is sync,
        - overrides the global-sync-timeout;
        - defaults to the global-sync-timeout if not provided.
    ";
  }
}

grouping is-async-command {
  description "is-async-command is to be supported by all manifest commands even if only
    supported as sync or async. In such cases, a must statement should be
    included to limit support to either sync or async.";
  leaf is-async {
    type boolean;
    default "true";
    description "command can be supported as either an async or sync command by a vendor.
      When supported as a sync command, the OWB-C will determine the success/failure
      of the command based on the RPC response instead of waiting for transient
      notifications from the device.";
  }
}

grouping transfer-command {
  description "transfer-command defines the common set of variables used by download-file
    and upload-file";
  leaf remote-filename {
    type string;
    mandatory true;
    description "See command for detailed description.";
  }
  leaf local-file-path {
    type string;

```

```

        mandatory true;
        description "See command for detailed description.";
    }
    uses timeout-command;
    uses is-async-command;
}

grouping file-command {
    description "file-command is used by all manifest files needing a filename";
    leaf filename {
        type string;
        description "filename is mandatory for delete-file; optional otherwise.
            See command for detailed description.";
    }
}

grouping command-reboot {
    description "command-reboot is used by manifest commands which result in a
        device restart.";
    leaf auto-reboot {
        type uint16;
        mandatory true;
        description "See command for detailed description.";
    }
}

grouping download-file-command {
    description "down-file-command";
    container download-file {
        when "../command = 'download-file'";
        uses transfer-command;
        description "Transfer a file from the SFTP server to the device.
            format: download-file remote-filename local-file-path [timeout]
            where
                remote-filename is the filename of the file to transfer on the SFTP
                server. The filename can include a relative path that represents the
                subdirectory structure of the vendor's software directory. This file
                (and optional path) must exist in the software release directory on
                the SFTP server.

                local-file-path is the local path and filename to transfer the file on
                the device.

                timeout - see timeout-command grouping for basic details;
                        if command is async,
                        - Receipt of an in-progress (version 2)
                        transfer-notification resets the timeout.

                Maps to the transfer rpc with
                    action = download
                    local-file-path = local-file-path
                    remote-file-path =
                        sftp://user:password@host[:port]/path/remote-filename

                The remote-file-path attribute on the transfer command would be
                constructed by the software download agent by appending the sftp URL
                (which includes username, password, host, port, and path to the
                software release directory) with the remote_filename.

                In the context of the transfer, remote is the SFTP server (e.g., located
                on the software download agent) and local is on the Open ROADM device.

                Expected notifications: transfer-notification
            ";
    }
}

grouping upload-file-command {
    description "upload-file-command";
    container upload-file {
        when "../command = 'upload-file'";

```

```

uses transfer-command;
description "Transfer a file from the device to the SFTP server.
format: upload-file remote-filename local-file-path [timeout]
where
    remote-filename is the filename of the file to receive the upload
    on the SFTP server. The filename can include a relative path that
    represents the subdirectory structure of the vendor's software
    directory.

    local-file-path is the local path and filename of the file on
    the device to be uploaded to the SFTP server. This file must exist on
    the device.

    timeout - see timeout-command grouping for basic details;
              if command is async,
                - Receipt of an in-progress (version 2)
                  transfer-notification resets the timeout.

Maps to the transfer rpc with
    action = upload
    local-file-path = local-file-path
    remote-file-path =
        sftp://user:password@host[:port]/path/remote-filename

The remote-file-path attribute on the transfer command would be
constructed by the software download agent by appending the sftp URL
(which includes username, password, host, port, and path to the
software release directory) with the remote_filename.

In the context of the transfer, remote is the SFTP server (e.g., located
on the software download agent) and local is on the Open ROADM device.

Expected notifications: transfer-notification
";
}
}

grouping delete-file-command {
description "delete-file-command";
container delete-file {
    when "../command = 'delete-file'";
    uses file-command {
        refine filename {
            mandatory true;
        }
    }
}
uses timeout-command;
uses is-async-command;
must "is-async != 'false'" {
    error-message "delete-file is only supported as sync command";
}
description "Delete a file from the device's file system.
format: delete-file filename [timeout]
where
    filename is the filename to be deleted from the device. The filename
    may include path information.

    timeout - overrides the global-sync-timeout; defaults to the
              global-sync-timeout if not provided.

Maps to the delete-file rpc:
    delete-file filename
";
}
}

grouping sw-stage-command {
description "sw-stage-command";
container sw-stage {
    when "../command = 'sw-stage'";
    uses file-command;

```

```

uses timeout-command;
uses is-async-command;
description "Stage a file in the device. The details of what a device does during
the staging operation is vendor specific. However, the vendor may
initiate additional file transfers from the SFTP server during the
staging operation. It is expected that the files will only be
transferred from the software release directory.

format: sw-stage [filename] [timeout]
where
  filename is the filename of the file to stage. If filename is not
  provided, the software download application will send the sw-stage
  command without a filename.

  timeout - overrides the global-async-timeout; defaults to the
            global-async-timeout if not provided.

Maps to the sw-stage rpc:
  sw-stage [filename]

Expected notifications: sw-stage-notification
";
}
}

grouping sw-activate-command {
  description "sw-activate-command";
  container sw-activate {
    when "../command = 'sw-activate'";
    leaf version {
      type string;
      mandatory true;
      description "Although version is optional in the sw-activate rpc, it is
        mandatory in the manifest file command.";
    }
    leaf validation-timer {
      type string;
      description "hh-mm-ss";
    }
  }
  uses timeout-command;
  uses command-reboot;
  uses is-async-command;
  must "is-async != 'true'" {
    error-message "sw-activate is only supported as async command";
  }
  description "Activate a software load in a device. The details of what a device does
during the activation phase is vendor specific. The device initiates
an automatic reboot as part of the activation.

format: sw-activate version [validation-timer] [timeout] auto-reboot
where:
  version: The version of software that is being activated. (The current
  YANG model indicates that version is optional; however, version should
  be a mandatory attribute of the sw-activate command in the manifest
  file).

  validation-timer: Validation timer setting for the software activation.
  Format is expected to be in the form HH-MM-SS per the YANG model. The
  software download application expects this format in order to treat
  00-00-00 and no validation timer as the same use case.

  timeout - overrides the global-async-timeout; defaults to the
            global-async-timeout if not provided. This timer begins as soon as the
            sw-activate processing begins. timeout must be greater than the
            auto-reboot time.

  auto-reboot: time (in seconds) to wait to for the device to reboot.
  This is the device restart time (e.g. the length of time from device
  comm loss until the device is ready for login). This timer begins when
  the controller detects the comm-loss from the device. If login is not

```

```

        successful when this timer expires, the sw-activate is failed.

        NOTE: if controller swdl application is not doing the login directly,
        the controller may need to augment the auto-reboot timer to account for
        the login time.

        Maps to the sw-activate rpc:
            sw-activate version [validationTimer]

        Expected notifications: sw-activate-notification
            When no validation timer (or validation-timer = 00-00-00), two
            notifications will be expected: one for activate, the other for
            commit. Otherwise, only the activate notification is expected.

        NOTE: the sw-activate-notifications (for activate) may be received
        before or after the reboot; it is assumed the sw-activate-notification
        (for commit) always occurs after the reboot. Any polling due to missed
        sw-activate-notifications (activate and/or commit) should not be done
        until after the reboot login; processing of sw-activate does not
        complete until after receipt of the notifications and the reboot login.
    ";
}
}

grouping db-backup-command {
    description "db-backup-command";
    container db-backup {
        when "../command = 'db-backup'";
        uses file-command;
        uses timeout-command;
        uses is-async-command;
        description "Perform a database backup on the device.

            format: db-backup [filename] [timeout]
            where
                filename is the filename of the backup file to be generated on the
                device. If filename is not provided, the database backup application
                will send the db-backup command without a filename. It's possible the
                filename will not be statically provided in the manifest file, but
                provided by the database backup application.

            timeout - see timeout-command grouping for basic details;
                     if command is async,

        Maps to the db-backup rpc:
            db-backup [filename]

        Expected notifications: db-backup-notification
    ";
}
}

grouping db-restore-command {
    description "db-restore-command";
    container db-restore {
        when "../command = 'db-restore'";
        uses file-command;
        leaf node-id-check {
            type string;
            default "true";
            description "Defined as an string here so that manifest file can parameterize
                the value for user input. __NODE-ID-CHECK is used for that purpose. Other valid
                values are true or false. Maps to a boolean value in the rpc invocation.";
        }
        uses timeout-command;
        uses is-async-command;
        description "Perform a database restore on the device.

            format: db-restore [filename] [node-id-check] [timeout]
            where
                filename is the filename of the file to be restored on the

```



```

device. If filename is not provided, the database restore application
will send the db-restore command without a filename. It's possible the
filename will not be statically provided in the manifest file, but
provided by the database restore application.

node-id-check is a boolean indicating whether nodeIDCheck is required.

timeout - see timeout-command grouping for basic details;
         if command is async,

Maps to the db-restore rpc:
  db-restore [filename] [nodeIDCheck]

Expected notifications: db-restore-notification
";
}
}

grouping db-activate-command {
  description "db-activate-command";
  container db-activate {
    when "../command = 'db-activate'";
    leaf rollback-timer {
      type string;
      description "hh-mm-ss";
    }
    uses timeout-command;
    uses command-reboot;
    uses is-async-command;
    must "is-async != 'true'" {
      error-message "db-activate is only supported as async command";
    }
  }
  description "Activate a database on a device. The details of what a device does
during the activation phase is vendor specific. The device initiates
an automatic reboot as part of the activation.

format: db-activate [rollback-timer] [timeout] auto-reboot
where:
  rollback-timer: Rollback timer setting for the database activation.
Format is expected to be in the form HH-MM-SS per the YANG model. The
database activation application expects this format in order to treat
00-00-00 and no validation timer as the same use case.

timeout - overrides the global-async-timeout; defaults to the
global-async-timeout if not provided. This timer begins as soon as the
db-activate processing begins. timeout must be greater than the
auto-reboot time.

auto-reboot: time (in seconds) to wait to for the device to reboot.
This is the device restart time (e.g. the length of time from device
comm loss until the device is ready for login). This timer begins when
the controller detects the comm-loss from the device. If login is not
successful when this timer expires, the db-activate is failed.

NOTE: if controller database application is not doing the login
directly, the controller may need to augment the auto-reboot timer to
account for the login time.

Maps to the db-activate rpc:
  db-activate [rollBackTimer]

Expected notifications: db-activate-notification
When no rollback timer (or rollback-timer = 00-00-00), two
notifications will be expected: one for activate, the other for
commit. Otherwise, only the activate notification is expected.

NOTE: the db-activate-notifications (for activate) may be received
before or after the reboot; it is assumed the db-activate-notification
(for commit) always occurs after the reboot. Any polling due to missed
db-activate-notifications (activate and/or commit) should not be done
until after the reboot login; processing of db-activate does not

```

```

        complete until after receipt of the notifications and the reboot login.
    ";
}
}
}

```

## 9.3 SAMPLE MANIFEST FILES

### 9.3.1 Example Software Download File

```

{
  "vendor": "Vendor-A",
  "model": "OpenROADM-Device",
  "sw-version": "2.0",
  "global-async-timeout": 600,
  "instruction-set": [
    {
      "index": 1,
      "sw-manifest-commands": {
        "sw-manifest-command": [
          {
            "command-order": 1,
            "command": "org-openroadm-manifest-file:download-file",
            "download-file": {
              "remote-filename": "OPENROADM-02001.PGM",
              "local-file-path": "/var/ftp/OPENROADM-02001.PGM",
              "timeout": 300,
              "is-async": "true"
            }
          },
          {
            "command-order": 2,
            "command": "org-openroadm-manifest-file:sw-stage",
            "sw-stage": {
              "filename": "/var/ftp/OPENROADM-02001.PGM",
              "timeout": 300,
              "is-async": "true"
            }
          },
          {
            "command-order": 3,
            "command": "org-openroadm-manifest-file:delete-file",
            "delete-file": {
              "filename": "/var/ftp/OPENROADM-02001.PGM",
              "is-async": "false"
            }
          },
          {
            "command-order": 4,
            "command": "org-openroadm-manifest-file:download-file",
            "download-file": {
              "remote-filename": "OPENROADM-02002.PGM",
              "local-file-path": "/var/ftp/OPENROADM-02002.PGM",
              "timeout": 300
            }
          },
          {
            "command-order": 5,
            "command": "org-openroadm-manifest-file:sw-stage",
            "sw-stage": {
              "filename": "/var/ftp/OPENROADM-02002.PGM",
              "timeout": 300
            }
          },
          {
            "command-order": 6,
            "command": "org-openroadm-manifest-file:delete-file",

```

```

        "delete-file": {
            "filename": "/var/ftp/OPENROADM-02002.PGM",
            "is-async": "false"
        }
    },
    {
        "command-order": 7,
        "command": "org-openroadm-manifest-file:sw-activate",
        "sw-activate": {
            "validation-timer": "01-00-00",
            "auto-reboot": 240,
            "version": "2.0",
            "timeout": 660,
            "is-async": "true"
        }
    }
]
}
}
]
}

```

### 9.3.2 Example Database Backup Manifest File

```

{
    "vendor": "Vendor-A",
    "model": "OpenROADM-Device",
    "sw-version": "x.y",
    "global-async-timeout": 900,
    "global-sync-timeout": 135,
    "db-backup-manifest-commands": {
        "db-backup-manifest-command": [
            {
                "command-order": 1,
                "command": "org-openroadm-manifest-file:db-backup",
                "db-backup": {
                    "filename": "__LOCAL-FILE-PATH",
                    "timeout": 180,
                    "is-async": "false"
                }
            },
            {
                "command-order": 2,
                "command": "org-openroadm-manifest-file:upload-file",
                "upload-file": {
                    "remote-filename": "__REMOTE-FILENAME",
                    "local-file-path": "__LOCAL-FILE-PATH",
                    "timeout": 240,
                    "is-async": "false"
                }
            },
            {
                "command-order": 3,
                "command": "org-openroadm-manifest-file:delete-file",
                "delete-file": {
                    "filename": "__LOCAL-FILE-PATH",
                    "is-async": "false"
                }
            }
        ]
    }
}

```

### 9.3.3 Example Database Restore Manifest File

```

{
  "vendor": "Vendor-A",
  "model": "OpenROADM-Device",
  "sw-version": "x.y",
  "global-async-timeout": 900,
  "global-sync-timeout": 135,
  "is-commit-db-activate-async": "false",
  "cancel-rollback-timer-async-timeout": 1000,
  "cancel-rollback-timer-sync-timeout": 140,
  "database-init-sync-timeout": 140,
  "db-restore-manifest-commands": {
    "db-restore-manifest-command": [
      {
        "command-order": 1,
        "command": "org-openroadm-manifest-file:download-file",
        "download-file": {
          "remote-filename": "__REMOTE-FILENAME",
          "local-file-path": "__LOCAL-FILE-PATH",
          "timeout": 240,
          "is-async": "false"
        }
      },
      {
        "command-order": 2,
        "command": "org-openroadm-manifest-file:db-restore",
        "db-restore": {
          "node-id-check": "__NODE-ID-CHECK",
          "filename": "__LOCAL-FILE-PATH",
          "timeout": 240,
          "is-async": "false"
        }
      },
      {
        "command-order": 3,
        "command": "org-openroadm-manifest-file:delete-file",
        "delete-file": {
          "timeout": 240,
          "filename": "__LOCAL-FILE-PATH",
          "is-async": "false"
        }
      },
      {
        "command-order": 4,
        "command": "org-openroadm-manifest-file:db-activate",
        "db-activate": {
          "rollback-timer": "01-00-00",
          "auto-reboot": 240,
          "timeout": 600
        }
      }
    ]
  }
}

```