



AppSel Code Provisioning

This chapter introduces Application Select (AppSel) code provisioning, a key feature for configuring the operating modes of optical modules. It details the benefits, operational mechanisms, and configuration steps for AppSel code provisioning.

Table 1: Feature History

Feature Name	Release Information	Feature Description
Application select code provisioning for QXP card	Cisco IOS XR Release 25.3.1	You can now configure and select application codes directly on a QSFP-DD module connected to a QXP-K9 card using the CLI. The pluggable modules supported is DP04QSDD-ULH-A1. This enhancement simplifies provisioning by allowing you to choose from advertised application modes such as 400ZR, OpenZR+, and others directly through the interface. Once selected, the host device activates the appropriate code, ensuring compatibility and streamlining configuration. CLI: The appsel simple code keyword is introduced in the controller optics command. The appsel detailed, appsel advertised, and appsel active keywords are introduced in the show controller optics command.

- [Application select code provisioning, on page 2](#)
- [Benefits of AppSel code provisioning, on page 3](#)
- [Configuration guidelines and restrictions for AppSel code provisioning, on page 3](#)

- [How AppSel code provisioning works, on page 4](#)
- [Configure an AppSel code on an optical module, on page 5](#)

Application select code provisioning

Application select code, also known as AppSel code, is a feature that:

- allows the host device to choose the operating mode of a QDD module
- allows you to configure the media code, and
- specifies how to configure the optical side of a module.

Common Management Interface Specification

The Common Management Interface Specification (CMIS) specifications set rules for how QDD modules work and how a host device sets them up. CMIS provides a consistent way for host devices, such as routers or switches, to communicate with and control optical modules, regardless of the module manufacturer.

Understanding AppSel codes and module operation modes

Each QDD module can operate in different modes, identified by a unique AppSel code. This code acts like a unique ID for each mode. Each mode has an application descriptor, which explains how the module handles data. It describes how signals are processed between the connections on the host side and the optical side.

The optical side is also known as the media side of the module. The AppSel code includes a media code, which tells the module how to set up its optical side. The host software uses this media code to control the module's optical interface. The host software also sets up other components, like the physical layer (PHY), SerDes, and MacPort, to complete the data path.

AppSel code storage and compatibility

Modules store a list of the AppSel codes they support in their EEPROM memory. This helps the host system or device know which settings the module can use. Some codes, like 400G-OIF-ZR and 400G-OpenZR+, are standard and follow industry rules. Other codes are custom, created by third-party vendors to provide users with more options on a single module. These custom codes may cause compatibility problems; for instance, Cisco routers may not function properly with modules using unrecognized custom codes.

Application mode selection

This feature allows users to select from all application modes advertised by the optical module. You can choose all application modes a module supports.

AppSel operation mode parameters

AppSel supports operation modes based on these parameters:

- Data rate, such as 100GE or 400GE
- Signal type
- Signal processing between host side and media side
- Optical configuration, based on the media code, which is part of the AppSel code.

Supported card and pluggables

In Release 25.3.1, AppSel is supported on the DP04QSDD-ULH-A1 pluggable connected to the QXP-K9 card.

Benefits of AppSel code provisioning

This section explains the benefits of provisioning AppSel code for flexible and efficient configuration of optical modules.

These are the benefits of provisioning AppSel code:

- Enables you to choose any application mode supported by the optical module for greater flexibility.
- Enables direct configuration of the NPU, PHY, and optics to match the selected application's datapath.
- Removes the need for Cisco IOS XR software to implement new proprietary modes for each vendor, simplifying software requirements.
- Allows you to directly specify the desired AppSel code, enabling straightforward configuration without requiring Cisco IOS XR software to interpret the mode.
- Eliminates the extra step of mapping custom vendor codes, reducing delays in supporting new module vendors.

Configuration guidelines and restrictions for AppSel code provisioning

This section explains how to properly configure and manage AppSel code provisioning for optical modules and describes its restrictions.

Configuration guidelines for AppSel code provisioning

You must follow these configuration guidelines for AppSel code provisioning.

- This feature is not enabled by default. You must configure it to take effect.
- The maximum number of applications supported depends on the module that is advertised in the AppSel list.
- For ULH pluggable, the AppSel code is a mandatory parameter. If it is not configured, a "Mandatory Configuration Missing" alarm is triggered and optics laser remains in off state.
- If you onfigure an invalid Appsel code during setup, a "Provisioning Failure" alarm will be raised.

How AppSel code provisioning works

Summary

The key components involved in the AppSel code provisioning are:

- **Optical side:** The optical side or the media side refers to the part of a pluggable module, such as a QDD or other optical transceiver that connects to and communicates with the optical network. It is the interface responsible for converting electrical signals, from the host device, into optical signals for transmission over fiber optic cables, and vice versa for receiving signals.
- **Host side:** The host side refers to the part of a pluggable module, such as a QDD or other optical transceiver that interfaces with the host device. A host device is typically a network device like a router, switch, or server. It is responsible for handling and processing electrical signals exchanged between the module and the host device.
- **Optics driver:** The entity that collects all supported application codes from the module. Users can see all supported application codes and select any one of the supported application codes.
- **Application codes:** The unique codes that represent operational modes of the optical module.
- **Optics Management Agent (MA):** The entity that sends the default configuration to the OSA driver if no user configuration is provided. It also manages fallback behavior when no application code is selected.

Workflow

These stages describe how the AppSel code provisioning process works.

1. AppSel code identifies the optical module modes when the modules are plugged into a router.
2. AppSel code points to an application descriptor. In this stage, the AppSel code acts as a sequence number for an application descriptor.
3. The application descriptor defines the configuration. In this stage, the application descriptor describes a functional transmission configuration, including signal processing between host lanes and media lanes.
4. The AppSel code that contains the media code configures the media side of the optical module.
5. The host software applies the media code to configure the optical interface of the optical module.
6. Based on the module's host side interface, you can configure PHY, NPU SerDes, and MacPort.
7. The host side and media side configurations complete, establishing the datapath.

Result

The host enables AppSel code provisioning as the data path is complete between the host side and media side. This ensures that the optical module operates correctly and efficiently in the desired mode, with proper coordination between the host side interface and the media side.

Configure an AppSel code on an optical module

Configure the AppSel code to enable the optical module to operate in a specific application mode, such as 400ZR or OpenZR+.

AppSel codes are advertised by the module and must be validated before configuration. This ensures compatibility between the host and the module.

Follow these steps to identify the AppSel codes available in a pluggable module and to configure them.

Procedure

Step 1 Identify the AppSel code that needs to be configured on a particular port from the list of available app codes.

Example:

```
RP/0/RP0/CPU0:ios#sh controllers optics 0/0/0/0 appsel advertised
Thu Aug 14 07:17:38.095 UTC
```

App-ID	Host-ID	Power	Media-ID	Standard
	Host	Power		
	Supported	Consumption (W)		
1	17	ETH 400GAUI-8 C2M (Annex	100	OpenROADM FLEXO-4e-DO-QP OpenROADM
	Yes	n/a		
2	15	ETH 200GAUI-4 C2M (Annex	100	OpenROADM FLEXO-4e-DO-QP OpenROADM
	No	n/a		
3	13	ETH 100GAUI-2 C2M (Annex	100	OpenROADM FLEXO-4e-DO-QP OpenROADM
	Yes	n/a		
4	17	ETH 400GAUI-8 C2M (Annex	216	FlexO 4e DPO 098Gbd Prop FlexO
	Yes	n/a		
5	15	ETH 200GAUI-4 C2M (Annex	216	FlexO 4e DPO 098Gbd Prop FlexO
	No	n/a		
6	13	ETH 100GAUI-2 C2M (Annex	216	FlexO 4e DPO 098Gbd Prop FlexO
	Yes	n/a		
7	17	ETH 400GAUI-8 C2M (Annex	201	FlexO 4e DPO 087Gbd Prop FlexO
	Yes	n/a		
8	15	ETH 200GAUI-4 C2M (Annex	201	FlexO 4e DPO 087Gbd Prop FlexO
	No	n/a		
9	13	ETH 100GAUI-2 C2M (Annex	201	FlexO 4e DPO 087Gbd Prop FlexO
	Yes	n/a		
10	17	ETH 400GAUI-8 C2M (Annex	194	FlexO 4e DPO 075Gbd Prop FlexO
	Yes	n/a		
11	15	ETH 200GAUI-4 C2M (Annex	194	FlexO 4e DPO 075Gbd Prop FlexO
	No	n/a		
12	13	ETH 100GAUI-2 C2M (Annex	194	FlexO 4e DPO 075Gbd Prop FlexO
	Yes	n/a		
13	17	ETH 400GAUI-8 C2M (Annex	192	FlexO 4e DPO 066Gbd Prop FlexO
	Yes	n/a		
14	15	ETH 200GAUI-4 C2M (Annex	192	FlexO 4e DPO 066Gbd Prop FlexO
	No	n/a		
15	13	ETH 100GAUI-2 C2M (Annex	192	FlexO 4e DPO 066Gbd Prop FlexO
	Yes	n/a		
16	17	ETH 400GAUI-8 C2M (Annex	198	OpenZR+ 400G 16QAM Propr OpenZR+
	Yes	n/a		
17	15	ETH 200GAUI-4 C2M (Annex	198	OpenZR+ 400G 16QAM Propr OpenZR+

```
RP/0/RP0/CPU0:ios#sh controllers optics 0/0/0/0 appsel detailed
Thu Aug 14 07:17:44.806 UTC
```

App-ID	Host-ID	Media-ID	Host Lane	Media Lane	Host Lane	Media Lane
Host			Count	Count	Assign	Assign
Supported						
1	17	100	8	1	0x1	0x1
Yes						
2	15	100	4	1	0x11	0x1
No						
3	13	100	2	1	0x55	0x1
Yes						
4	17	216	8	1	0x1	0x1
Yes						
5	15	216	4	1	0x11	0x1
No						
6	13	216	2	1	0x55	0x1
Yes						
7	17	201	8	1	0x1	0x1
Yes						
8	15	201	4	1	0x11	0x1
No						
9	13	201	2	1	0x55	0x1
Yes						
10	17	194	8	1	0x1	0x1
Yes						
11	15	194	4	1	0x11	0x1
No						
12	13	194	2	1	0x55	0x1
Yes						
13	17	192	8	1	0x1	0x1
Yes						
14	15	192	4	1	0x11	0x1
No						
15	13	192	2	1	0x55	0x1

Yes								
16	17	198	8	1	0x1	0x1		
Yes								
17	15	198	4	1	0x11	0x1		
No								
18	13	198	2	1	0x55	0x1		
Yes								
19	17	70	8	1	0x1	0x1		
Yes								
20	15	70	4	1	0x11	0x1		
No								
21	13	70	2	1	0x55	0x1		
Yes								
22	17	54	8	1	0x1	0x1		
Yes								
23	15	54	4	1	0x11	0x1		
No								
24	13	54	2	1	0x55	0x1		
Yes								
25	60	220	2	1	0x55	0x1		
No								
26	60	202	2	1	0x55	0x1		
No								
27	60	196	2	1	0x55	0x1		
No								
28	60	193	2	1	0x55	0x1		
No								

Step 2 Configure the identified AppSel code on the optics interface.

Example:

```
RP/0/RP0/CPU0:ios#configure
Wed Mar 27 14:12:49.932 UTC
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/0
RP/0/RP0/CPU0:ios((config-Optics))#appsel simple code 13
RP/0/RP0/CPU0:ios((config-Optics))#commit
```

Step 3 Verify the AppSel code configured on the optics.

Example:

```
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#sh controllers optics 0/0/0/0 appsel active
Thu Aug 14 07:17:50.861 UTC
```

```
Instance          :1
App-ID            :13
Host-ID           :17  ETH 400GAUI-8 C2M (Annex 120E)
Media-ID          :192  FlexO 4e DPO 066Gbd Proprietary
Host Lane Count   :8
Media Lane Count  :1
Host Lane Assign  :0x1
Media Lane Assign :0x1
```

The optical module operates in the selected application mode, ensuring compatibility and optimal performance.

What to do next

- Monitor the interface status and confirm the active AppSel code.

- Ensure that alarms are cleared, and the interface is operational.